Universidade Estadual de Campinas

Faculdade de Engenharia Elétrica e Computação

Departamento de Comunicações

Criação de Modelos Organizacionais para Cidades Digitais Baseadas em uma Arquitetura Peer-to-Peer

Autor: Eder Ignatowicz

Orientador: Leonardo de Souza Mendes

Trabalho apresentado à Faculdade de Engenharia Elétrica e de Computação da UNICAMP como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática

Comissão Examinadora

Gean Davis Breda - Unicamp

Leonardo de Souza Mendes - Unicamp

Rodolfo Miranda de Barros – UEL

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Ig5c

Ignatowicz, Eder
Criação de modelos organizacionais para cidades digitais baseadas em uma arquitetura peer-to-peer / Eder Ignatowicz. --Campinas, SP: [s.n.], 2009.

Orientador: Leonardo de Souza Mendes.
Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Tecnologia da informação. 2. Vida comunitaria. 3. Redes de computação - Aspectos sociais. 4. Sociedade da informação. I. Mendes, Leonardo de Souza. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Building organization models for digital cities peer-to-peer based
Palavras-chave em Inglês: Communication in technology, Community life, Computer networks - Social aspects, Information society
Área de concentração: 2009
Titulação: Meste em Engenharia Elétrica
Banca examinadora: Gean Davis Breda , Leonardo de Souza Mendes , Rodolfo Miranda de Barros
Data da defesa: 11/11/2009
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Éder Ignatowicz

Data da Defesa: 11 de novembro de 2009

Título da Tese: "Criação de Modelos Organizacionais para Cidades Digitais Baseadas em uma Arquitetura Peer-to-Peer"

Prof. Dr. Leonardo de Souza Mendes (Presidente): _____

Prof. Dr. Rodolfo Miranda de Barros: _____

Dr. Gean Davis Breda: _____

Agradecimentos

Inicialmente, gostaria de agradecer a Deus por ter me acompanhado em todos os momentos da minha vida. Agradeço também a Deus por ter me proporcionado algo que tantos anseiam neste país, cursar uma faculdade e pós graduação pública e de qualidade.

Ao meus pais, Pedro e Eunice, por serem meus maiores exemplos e a base de tudo o que sou, meus anjos que me oferecem apoio irrestrito em todos os sonhos da minha vida.

Aos meus avós, irmãos (Pedro e Luciano), cunhadas(Márcia e Katia) e as sobrinhas mais lindas e carinhosas que alguém poderia sonhar (Alexia, Audrey e Laura). Pelo carinho, apoio e conselhos em todas as etapas da minha vida.

A minha namorada e companheira Ana Carolina Rodrigues de Moraes Garcia, pelo amor, carinho, afeito e compreensão em todos os momentos da minha vida.

Aos meus INCRÍVEIS amigos Diogo “Didio”, Douglas “Doug”, Liniquer “Lilão”, Guilherme “Jonny”, Tiago “Cabeça”, Osman “Oz”, Daniel “Danzão”, Fábio Campos “Líder”, Fernanda Takada “Fer”, Malu, Lutch “Cachaça”, Fernando Bertasso “Nandinho”, Reury, Ricardo “Cabelo”, Gustavo “Tato”, Sérgio “Don” e Marcelo Ventrilho “Broz”, Alex, Keyla, Iuri e Paulinho “Geo”.

Ao professor Leonardo de Souza Mendes, professor que desde minha chegada a Campinas me acolheu, apoiou e apostou em meu potencial, pela orientação e inspiração no decorrer destes ótimos anos.

Aos amigos André Panhan e Rodrigo Miani pelo apoio técnico e pelas inúmeras considerações feitas no decorrer deste trabalho.

Agradeço ao glorioso São Paulo Futebol Clube pelas vitórias, conquistas e imensas alegrias ao longo desta última década. 6-3-3

A todos os companheiros do LarCom(Laboratório de Comunicações e Redes de Computadores) pelos momentos de trabalho aliados a descontração e amizade.

Por fim, agradeço a todos os amigos, colegas e familiares não mencionados mas que colaboraram com minha formação pessoal e profissional, que de certa forma contribuiriam para que este trabalho se concretizasse.

Dedico este trabalho
À minha família;

Resumo

Guiado pelo crescente número de projetos de cidades digitais, o objetivo principal deste trabalho é propor uma arquitetura e um modelo organizacional de intercâmbio de informações em uma cidade digital construindo uma infra-estrutura de disponibilização e consumo de serviços pelos cidadãos. Este modelo tem como objetivo promover o acesso do cidadão aos serviços da cidade digital e prover meios de fácil e ampla disponibilização e divulgação destes serviços a iniciativa pública e privada por meio da arquitetura apresentada neste trabalho. Desta forma, podemos garantir que a informação ou serviço possa ser rapidamente localizada e transmitida entre os setores públicos e privado, criando assim um impulso na economia local. Este modelo é validado por meio do desenvolvimento de um portal comunitário construído sobre uma rede peer-to peer (P2P) e baseado no modelo MVC (Model-View-Controller) modificado para quatro camadas. Este portal utiliza um *middleware* para interconectar os mais variados segmentos da sociedade.

Palavras-chave: Portal Relacional Comunitário, Sistemas Web, Cidades Digitais, Redes Metropolitanas de Acesso Aberto, MVC.

Abstract

Support by the growing of digital cities projects, the main goal of this work is to propose an architecture and an organizational model of information exchange in a digital city building an infrastructure of deploying and consumption of services by citizens. This model aim to promote citizen access to city services and provide easy and wide digital media availability and dissemination of these services to public and private initiative through the architecture presented. So we can ensure that the information or service can be quickly located and exchanged between the public and private sectors, thus creating a rise in the local economy. This model is validated through the development of a Community Portal built on a peer-to peer (P2P) and based on the MVC (Model-View-Controller) modified(four layers). This website uses a middleware to interconnect the different segments of society.

Key-words: Community portals, web-based system, Digital Cities, Open Access Metropolitan Networks, MVC.

Sumário

1 Lista de Figuras ix
Lista de Tabelas x
Glossário xi
1 Introdução 1
1.1 Modelo organizacional 3
1.2 Contribuições da dissertação 4
1.3 Estrutura da dissertação 5
2 Definições Básicas 6
2.1 Cidades Digitais 6
2.2 Portal 7
2.3 Portal Relacional Comunitário 9
2.4 MAN – Metropolitan Area Network 9
2.5 Open MAN 10
2.6 Middleware 12
2.7 Peer-to-peer 13
2.7.1 Jxta 17
2.8 Web Semântica 19
3 Arquitetura de Cidades Digitais 24
3.1.1 Interconexão 25
3.1.2 Segurança 26
3.1.3 Meios de acesso 28
3.1.4 Organização 28
3.1.5 Benefícios da utilização da arquitetura proposta 29
3.1.6 Middleware 31
4 Portal Relacional Comunitário 36
4.1 Motivação 36
4.2 Cenário de Exemplo 37
4.3 Arquitetura do Portal 39
4.3.1 Model (Negócios) 40
4.3.2 View (Apresentação) 42
4.3.3 Controller (Aplicação) 42
4.3.4 Data Layer(Camada de Dados) 42
4.4 Comunicação com outros sistemas 43
4.5 Padrões definidos 44
5 Modelagem e implementação do portal relacional comunitário 50
5.1 Visão de caso de uso 50
5.2 Visão Lógica 52
5.2.1 Diagramas de pacotes 53
5.3 Visão de Distribuição 56
5.4 Visão dos Dados 56
5.5 Visão de Implementação 59
5.6 Testes 62
6 Conclusões e Trabalhos Futuros 65
7 Bibliografia 67

1 Lista de Figuras

Figura 1 Cidades Digitais 2
Figura 2 - Organização das cidades digitais 2
Figura 3 - Prédios Públicos - Pedreira 12
Figura 4 Arquitetura P2P Centralizada[68] 16
Figura 5 Arquitetura P2P Descentralizadas e desestruturadas 16
Figura 6 - Camadas JXTA[68] 18
Figura 7 - Portal de Serviços de Zaragoza [48] 21
Figura 8 - Portal de Asturias 22
Figura 9 - Segmentos de uma cidade digital 24
Figura 10 - Arquitetura para cidades digitais 30
Figura 11 - Middleware de uma cidade digital [12] 34
Figura 12 - Arquitetura de quatro camadas do Portal Relacional Comunitário [12] 40
Figura 13 – Arquitetura de conexão do portal relacional comunitário 44
Figura 14 - Xml de busca de categorias 46
Figura 15 - Xml de cadastro de produtos 47
Figura 16 – Xml de Transferência de usuário 48
Figura 17 - Xml de Requisição de informações do usuário 48
Figura 18 - Xml de informações do usuário 49
Figura 19 - Componentes de um caso de uso 51
Figura 20 - Diagrama de Caso de Uso (Usuário) 51
Figura 21 - Diagrama de Caso de Uso (Provedor) 52
Figura 22 - Diagrama de Caso de Uso (Portal) 52
Figura 23 - Diagrama de Componentes 53
Figura 24 - Diagrama de Pacotes (Camada de Apresentação) 54
Figura 25- Diagrama de Pacotes (Camada de Aplicação) 54
Figura 26 - Diagrama de Pacotes (Camada de Negócios) 55
Figura 27 - Diagrama de Pacotes (Camada de Dados) 55
Figura 28 - Visão de distribuição 56
Figura 29 – Modelo Relacional 57
Figura 30 - Mapeamento JPA da tabela Categorias 59
Figura 31 - Diagrama de Seqüência - Buscar Categoria 60
Figura 32 - Buscar Categoria 60
Figura 33 – Login 61
Figura 34 - Tela inicial 61
Figura 35 - Buscar Serviço 62
Figura 36 - Classe de testes - BuscarCategoriaHelper 64

Lista de Tabelas

Tabela 1 - Geração dos portais 8

Glossário

API – Application Programming Interface

B2B - Business to Business

B2C - Business-to-consumer

B2G - Business to Government

COBIT - Control Objectives for Information Technology

C2C - Consumer to Consumer

DHT - Distributed hash table

DNS - Domain Name System

EJB – Enterprise Java Bean

G2C - Government to Citizen

G2G - Government to Government

LAN - Local Area Network

IP - Internet Protocolo

ISO – International Standarts Organization

ITIL - Information Technology Infrastructure Library

JPA – Java Persistence API

JTXA - Juxtapose

MAN – Metropolitan Area Network

MVC - Model-View-Controller

ORM – Object Relational Mapping

PIB – Produto Interno Bruto

P2P - Peer-to peer

T.I. - Tecnologia da Informação

TCI - Tecnologias de Informação e Comunicação

UML - Unified Modelling Language

URI - Uniform Resource Identifier

VoD - Video on demand

VoIP - Voz sobre IP

WAN - Wide Area Network

WS - Web Semântica

XML - eXtensible Markup Language

Trabalhos Publicados pelo Autor

1. Panhan, A. M., Ignatowicz E., Mendes, L. S., Community portals for architecture-based middleware P2P on IEEE - LATINCOM 2009.

1 Introdução

Segundo Michaelis [1], cidade pode ser definida como um centro urbano onde estão geralmente localizadas as casas comercialmente mais importantes. No Brasil, o termo cidade também pode ser definido como sede de município, um aglomerado permanente, relativamente grande e denso, de indivíduos socialmente heterogêneos.

De acordo com Silva [2] a cidade, ainda, pode ser entendida como um espaço social onde todos moradores teriam, a priori, o direito de ir e vir, de compartilhar a cultura, a riqueza, os bens de serviços, desfrutar do conhecimento coletivo, direito ao trabalho e a participação nas decisões do uso dos espaços da cidade.

Com as recentes evoluções tecnológicas alcançadas principalmente pela microeletrônica e informática, a definição de cidades e relações sociais que nela são desempenhadas ganham uma nova visão baseadas no âmbito do ciberespaço. Destas evoluções emerge o conceito de Cidade Digital.

Segundo Ishida [3], o termo Cidade Digital, se refere a uma comunidade conectada que combina uma infra-estrutura de telecomunicações flexível, com uma rede de serviços baseada em padrões abertos e que fornece serviços inovadores que atendem toda a sociedade. Contudo, a definição do que é uma cidade digital é divergente na literatura. O capítulo 2 apresentará a visão de diversos autores a respeito do conceito de cidade digital.

Desta forma, embora a expressão “Cidades Digitais” (Figura 1) não possua uma definição precisa, desde os anos 90 [4] tem-se apresentado um crescente número de projetos de cidades digitais visando a criação de ambientes de prestação de serviços eletrônicos aos cidadãos.

Estes projetos como, por exemplo, as Infovias Municipais [5], geraram importantes avanços tecnológicos nas áreas de telecomunicações e informática. Dentre eles podemos citar novos modelos e arquiteturas de gestão de recursos, organização e transporte de informações [5][6]. Segundo Sttot [7], estas inovações modificarão de forma qualitativa as geografias sociais.

Figura 1 Cidades Digitais

A construção de cidades digitais envolvem a análise de uma gama de setores da sociedade [7][8][9]. A Figura 2, apresenta os principais aspectos de enfoque na construção de uma cidade digital.

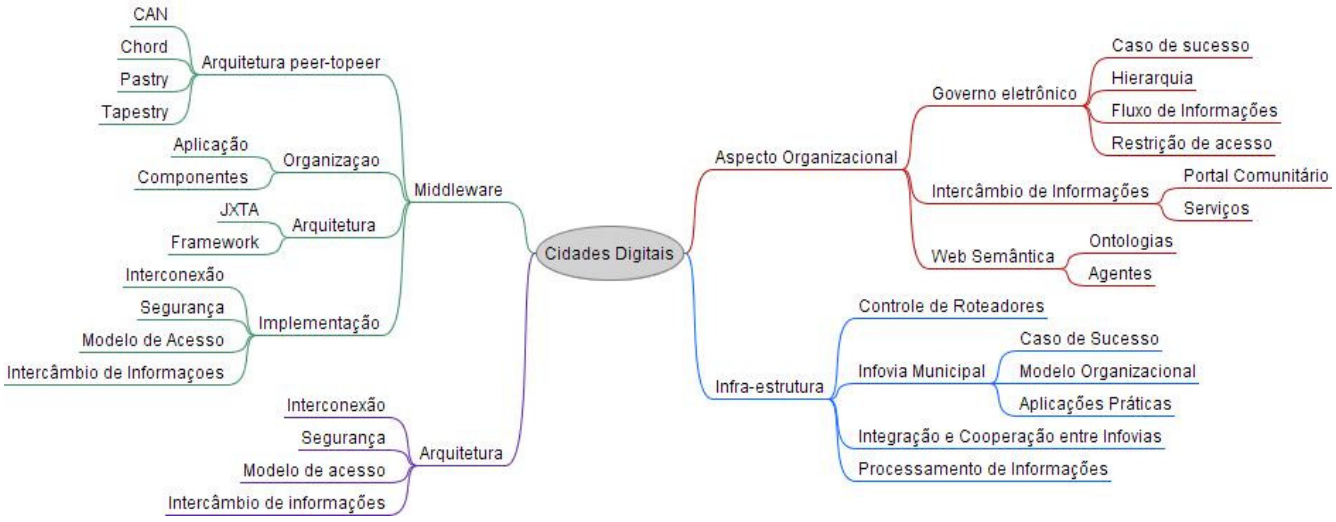

Figura 2 - Organização das cidades digitais

Apresentado como o mapa mental na Figura 2, pode-se dividir em quatro áreas os requisitos para construção de uma cidade digital:

1. **Infra Estrutura:** Lida com questões relativas à interconexão de rede entre os mais variados segmentos da sociedade em uma rede de dados;
2. **Middleware:** Fornece mecanismos de interconexão lógica entre os diversos serviços oferecidos pelos cidadãos em uma cidade digital;
3. **Arquitetura:** Estabelece requisitos básicos para interconexão, aspectos relativos à segurança de informação, requisitos para intercâmbio de informação e modelo de acesso aos recursos de uma cidade digital;
4. **Aspecto Organizacional:** Descreve um modelo organizacional de disponibilização de serviços através de uma infra-estrutura de intercâmbio de informações. Padroniza mecanismos para troca de mensagens entre os diversos atores da cidade digital. Lida também com questões relativa a semântica dos dados.

Como a motivação deste trabalho é apresentar um modelo capaz de criar uma infra-estrutura de organização de uma cidade digital, questões relativas à infra-estrutura física, middleware e arquitetura serão tratadas brevemente nos capítulos 2 e 3. Maiores detalhes destes aspectos podem ser encontrados em [5] e [10].

1.1 *Modelo organizacional*

Guiado pelo crescente número destes projetos de cidades digitais, o objetivo principal deste trabalho é propor um modelo organizacional de intercâmbio de informações em uma cidade digital. Desta forma será criada uma infra-estrutura de disponibilização e consumo de serviços pelos cidadãos. Ele é construído basicamente através de estruturas XML(eXtensible Markup Language) criadas com o intuito de padronizar a troca de mensagens entre os atores da Cidade Digital.

Este modelo tem como objetivo facilitar o acesso do cidadão aos serviços da cidade digital e prover meios de fácil e ampla disponibilização e divulgação destes serviços a iniciativa pública e privada, através da definição da arquitetura apresentada neste trabalho.

Desta forma podemos garantir que a informação ou serviço possa ser rapidamente localizada e transmitida entre os setores públicos e privado, possibilitando assim um grande impulso na economia local.

Este modelo é validado através do desenvolvimento de um portal relacional comunitário construído sobre uma rede *peer-to peer* (P2P) e baseado no modelo MVC (*Model-View-Controller*) modificado de quatro camadas [11]. Este portal utiliza um *middleware* para interconectar os mais variados segmentos da sociedade.

Por ser projetado como uma estrutura básica para o planejamento estratégico tanto da iniciativa pública quanto privada em diversos cenários de negócios (B2B, B2C, B2G, G2G, G2C e C2C) o portal é projetado para lidar com questões relativas à política de acesso, privacidade, segurança e apresentar ao cidadão uma *interface* amigável e intuitiva.

1.2 Contribuições da dissertação

O objetivo principal deste trabalho é propor um modelo de intercâmbio de informações em uma cidade digital. Este modelo possibilitará a criação de uma infra-estrutura de organização, disponibilização e consumo de serviços pelos cidadãos.

Este modelo será validado através da construção de uma arquitetura e um portal relacional comunitário possibilitando aos mais diversos segmentos da sociedade compartilhar serviços através da Cidade Digital. Resumidamente, as principais contribuições da dissertação são:

- Discussão e definição conceitual e arquitetural de uma cidade digital;
- Proposta de uma arquitetura para cidades digitais;
- Definição de um modelo de intercâmbio de informações em uma cidade digital;
- Proposta de uma arquitetura para possibilitar este intercâmbio de informações;
- Criação de padrões de comunicação entre provedores de serviço e o portal relacional comunitário;
- Proposta de uma plataforma de baixo custo (portal comunitário) para oferta e consumo de serviços a toda a comunidade da cidade digital;

1.3 Estrutura da dissertação

Os capítulos desta dissertação estão dispostos da forma a seguir.

O Capítulo 2 apresenta um levantamento bibliográfico de alguns dos temas principais abordados na dissertação.

O Capítulo 3 apresenta a conceituação geral de cidades digitais. Este capítulo também detalha a arquitetura e o middleware proposto em [10] e [12] e detalhados neste trabalho.

O Capítulo 4 descreve a conceituação geral do Portal Relacional Comunitário. Apresenta sua motivação em seguida um estudo de caso para compreensão do problema. Em seguida descreve a arquitetura proposta, a forma de realizar comunicação com outros sistemas e os padrões definidos de comunicação.

No capítulo 5, é apresentado o projeto e desenvolvimento do protótipo do Portal Relacional Comunitário. Este portal é um protótipo construído para validar o modelo proposto por este trabalho.

No Capítulo 6 é sumarizado as contribuições obtidas e apresentada propostas de possíveis trabalhos futuros.

2 Definições Básicas

Antes de darmos seqüência com o trabalho, faz-se necessária a discussão prévia de alguns temas abordados nesta dissertação.

2.1 Cidades Digitais

Também conhecida como Cidade Virtual, Cidade Inteligente, entre outros, uma cidade digital é definida por Silva [2] como um sistema de pessoas e instituições conectadas por uma infra-estrutura de comunicação digital (a Internet) que tem como referência uma cidade real cujos propósitos variam e podem incluir um ou mais dos seguintes objetivos:

1. Criar um espaço de manifestação política e cultural das pessoas e grupos;
2. Criar um canal de comunicação entre as pessoas e grupos;
3. Criar canais de comunicação e negociação entre a administração municipal e os cidadãos;
4. Favorecer uma maior identificação dos moradores e visitantes com a cidade referência;
5. Criar um acervo de informações das mais variadas espécies e de fácil acesso sobre a cidade referência.

Ishida [3] explica que o termo Cidade Digital refere-se a uma comunidade conectada que combina uma infra-estrutura de telecomunicações flexível, com uma rede de serviços baseada em padrões abertos e que fornece serviços inovadores que atendem toda a sociedade.

Segundo Lemos [13] entende-se por Cidade Digital, a criação de infra-estrutura, serviços e acesso público em uma determinada área urbana para o uso das novas tecnologias e redes telemáticas. O objetivo é criar interfaces entre o espaço eletrônico e o espaço físico por meio de oferecimento de teleportos, telecentros, quiosques multimídia e áreas de acesso e serviços.

Barletta [14] define Cidades Digitais como uma representação de modelos que existem no mundo real.

"... É uma nova resposta social que se baseia nos desenvolvimentos tecnológicos que estão de uma forma ou de outra, alterando nosso comportamento. São cidades que se constituem através de interfaces criadas graças às facilidades que oferecem o desenvolvimento de software especializado para se trabalhar na Internet". "... É uma forma diferente de aproximar- se a uma cidade real". O autor também cita que "... uma Cidade Digital é a representação de uma nova forma de se tratar o conhecimento".

Cidades digitais, segundo Grahan [15], podem ser vistas como ferramentas desenvolvidas para aprimorar a comunicação entre cidadãos e o governo local, estimulando e facilitando o acesso a cidadãos aos serviços públicos.

"... Cidades Virtuais são espaços eletrônicos, em geral com base na World Wide Web, que foram desenvolvidos para interligar, de forma explícita, as agendas de desenvolvimento de cada cidade. Tais Cidades Virtuais estão funcionando como ferramenta política para uma variedade de planos e objetivos urbanos: marketing urbano global, estímulo ao turismo de negócios e de consumo, melhoria das comunicações entre os cidadãos e os governos locais, aumento da competitividade das empresas locais, maior integração das economias locais e o renascimento do civismo e da cultura local."

Sendo assim, embora o principal fator motivador da construção de uma cidade digital seja o aprimoramento entre as relações entre o cidadão e o governo local [5][15] também se percebe uma cidade digital como um espaço onde o cidadão pode divulgar suas idéias, negócios e realizar manifestações político-culturais das mais variadas formas possíveis.

2.2 Portal

Segundo Dias [16] um portal é um site na internet que funciona como centro aglomerador e distribuidor de conteúdo para uma série de outros sites ou subsites dentro, e também fora, do domínio ou subdomínio da empresa gestora do portal.

Na década de noventa, o que hoje conhecemos como um portal era conhecido como uma máquina de busca, cujo objetivo era facilitar o acesso as informações contidas em documentos espalhados na Internet. Inicialmente, as máquinas de busca possibilitavam ao usuário da Internet localizar documentos a partir de pesquisas e navegação associativa entre links. Para reduzir ainda mais o tempo de busca na Internet e auxiliar os usuários menos

experientes, vários sites de busca incluíram categorias, isto é, passaram a filtrar sites e documentos em grupos pré-configurados de acordo com seu conteúdo - esportes, meteorologia, turismo, finanças, notícias, cultura etc. O passo seguinte foi a integração de outras funções, como, por exemplo, as comunidades virtuais e suas listas de discussão, chats em tempo real, possibilidade de personalização dos sites de busca e acesso a conteúdos especializados e comerciais. Essa nova concepção de máquina de busca é que passou a ser chamada de portal [16].

De acordo com Shilakes [17] portais são aplicativos que permitem aos usuários uma única via de acesso à informação personalizada necessária para a tomada de decisões.

Segundo Eckerson, desde meados de 1990, os portais passaram por três gerações diferentes apresentadas na Tabela 1: [16]

Geração	Características
Primeira	Máquina de busca, com catálogo hierárquico de conteúdo da web. Cada entrada do catálogo contém uma descrição do conteúdo e um link.
Segunda	O usuário, por meio de um identificador e uma senha, pode criar uma visão personalizada do conteúdo do portal, conhecida como "Página Pessoal". Esta visão mostra apenas as categorias que interessam a cada usuário
Terceira	O portal incorpora serviços diversos, tais como correio eletrônico, chat, listas de discussão, cotação da bolsa, comércio eletrônico, etc. Permitindo ao usuário interagir com o portal e com seu provedor de conteúdo

Tabela 1 - Geração dos portais

Podemos acrescentar a definição de Eckerson[16] o conceito de portal de Web 2.0. A web 2.0 é uma mudança do paradigma da Internet. Esta mudança faz-se por meio da visualização da Internet como uma plataforma. Esta plataforma pode ser utilizada para o desenvolvimento de aplicativos que aproveitem todos os efeitos da Internet aliados à facilidades antes só encontradas em sistemas desktop.

Desta forma um portal é uma porta de entrada a uma rede global. Grande parte dos usuários define seus próximos passos de navegação na Web de acordo com o conteúdo do portal acessado inicialmente. Um portal também se caracteriza por possuir um grande

número de usuários. Desta forma, ser reconhecido como um portal está diretamente relacionado com a capacidade de este atrair audiência. [16]

Basicamente podemos dividir os portais em duas categorias. Portais que oferecem conteúdo e serviços genéricos e portais que possuem foco em um nicho específico como turismo, jogos, esportes ou serviços.

2.3 Portal Relacional Comunitário

Por tratar-se de web sites que possuem diversas funcionalidades e utilizam tecnologias de ponta, a construção de um portal que leve em consideração fatores como política de acesso, privacidade, segurança e facilidade de uso pode ser dispendiosa. Este alto custo muitas vezes impossibilita ao cidadão comum oferecer seus serviços na Internet.

Desta forma este trabalho propõe a construção de um portal de terceira geração (utilizando conceitos web 2.0), comum a todos os cidadãos da cidade digital com o intuito de validar o modelo organizacional proposto. Com a utilização deste modelo, todos os segmentos da sociedade poderão interagir e compartilhar serviços. Desta forma será realizada uma racionalização de investimentos em T.I., por meio do compartilhamento, resumo e intercâmbio de recursos tecnológicos.

Logo podemos definir um portal relacional comunitário como um portal de serviços que almeja facilitar a interação entre diversos segmentos da sociedade. Ele se apresenta como uma ferramenta ao usuário para consumir e prover serviços sem necessitar de um grande aparato tecnológico, apenas utilizando um browser de Internet.

2.4 MAN – Metropolitan Area Network

Em 1998, Mollenauer [18] definiu uma Metropolitan Area Network (MAN) como uma evolução natural do conceito da LAN (Local Area Network). Ele cita que uma típica MAN conecta diversos prédios de uma corporação com múltiplos pontos de conexão em uma área metropolitana.

Ming[19] define uma (MAN) como uma grande rede de computadores que abrange uma área metropolitana ou campus universitário. O seu âmbito geográfico situa-se entre uma LAN e WAN (Wide Area Network).

Uma rede MAN também pode ser vista como uma rede de interconexão entre LAN/LAN ou LAN/WAN para sistemas de comunicação pública, em uma área geográfica metropolitana. [20]

Segundo o IEEE [21] uma MAN é uma rede otimizada para uma área geográfica maior do que uma LAN (Local Area Network), abrangendo vários blocos de edifícios em cidades inteiras. A MAN pode ser propriedade de uma única organização, mas normalmente é utilizada por muitos indivíduos e organizações. MAN também podem ser exploradas como serviços de utilidade pública. Freqüentemente irão fornecer os meios para interconexão de redes locais.

Uma MAN pode ser caracterizada como uma rede que possui alta taxa de transferência de dados, alta confiança e um baixo nível de perda de pacotes[22]. Ela é composta basicamente das mesmas tecnologias e protocolos utilizados em uma LAN como, por exemplo, o protocolo TCP/IP.

2.5 *Open MAN*

Redes metropolitanas abertas (Open MAN) desempenham um papel de extrema importância na sociedade atual. Esta relevância é justificada pela capacidade deste modelo de rede prover acesso irrestrito a informação e conhecimento. O livre acesso a estes fatores apresenta-se como fator primordial ao aprimoramento dos processos produtivos e inovação, peças chave ao desenvolvimento econômico e social.

Podemos definir um Open MAN [5] como uma rede convergente multimídia que provê acesso universal para toda a população de uma cidade. Por tratar-se de uma definição abrangente, diversas soluções de rede se enquadram nesta categoria. No entanto, segundo Mendes [5], o estado da arte do desenvolvimento de redes TCP / IP amparado por soluções ópticas e sem fio Ethernet (Gigabit Ethernet, 10G Ethernet, IEEE802.11x, IEEE802.16x, etc), oferecem um ambiente único genérico que pode ser considerado o melhor candidato para tornar-se a solução de escolha para uma única rede convergente e universal.

Open MAN também pode ser vista como uma MAN comunitária criada com o intuito de prestação de serviços de interesse comum a uma cidade ou região.

Como exemplo de Open MAN podemos citar a Rede Metropolitana de Acesso Aberto de Pedreira. Esta Open MAN é um projeto entre o LaRCom-Unicamp (Laboratório de Redes de Comunicação da Unicamp) e a prefeitura da cidade. Os estudos para a implantação da rede começaram em 2005 e sua inauguração se deu em junho de 2007. Atualmente, a infra-estrutura de rede da Infovia de Pedreira é híbrida, formada por um backbone óptico que interliga diversos pontos no centro da cidade e também por aproximadamente 35 células de rádio espalhados pelo município. Todos estes pontos de rádio são utilizados para oferecer Internet de graça à população. Além das residências, os principais prédios públicos da cidade de Pedreira estão conectados a Open MAN. A Figura 3 mostra a disposição dos prédios públicos e a respectiva tecnologia de interconexão de rede. [23]

Os serviços atualmente disponibilizados e que trafegam sobre a Infovia são: distribuição de Internet, aplicações de *e-Government*, Voz sobre IP (VoIP), E-mail, Rede de Sensores e Câmeras IP para a segurança pública.

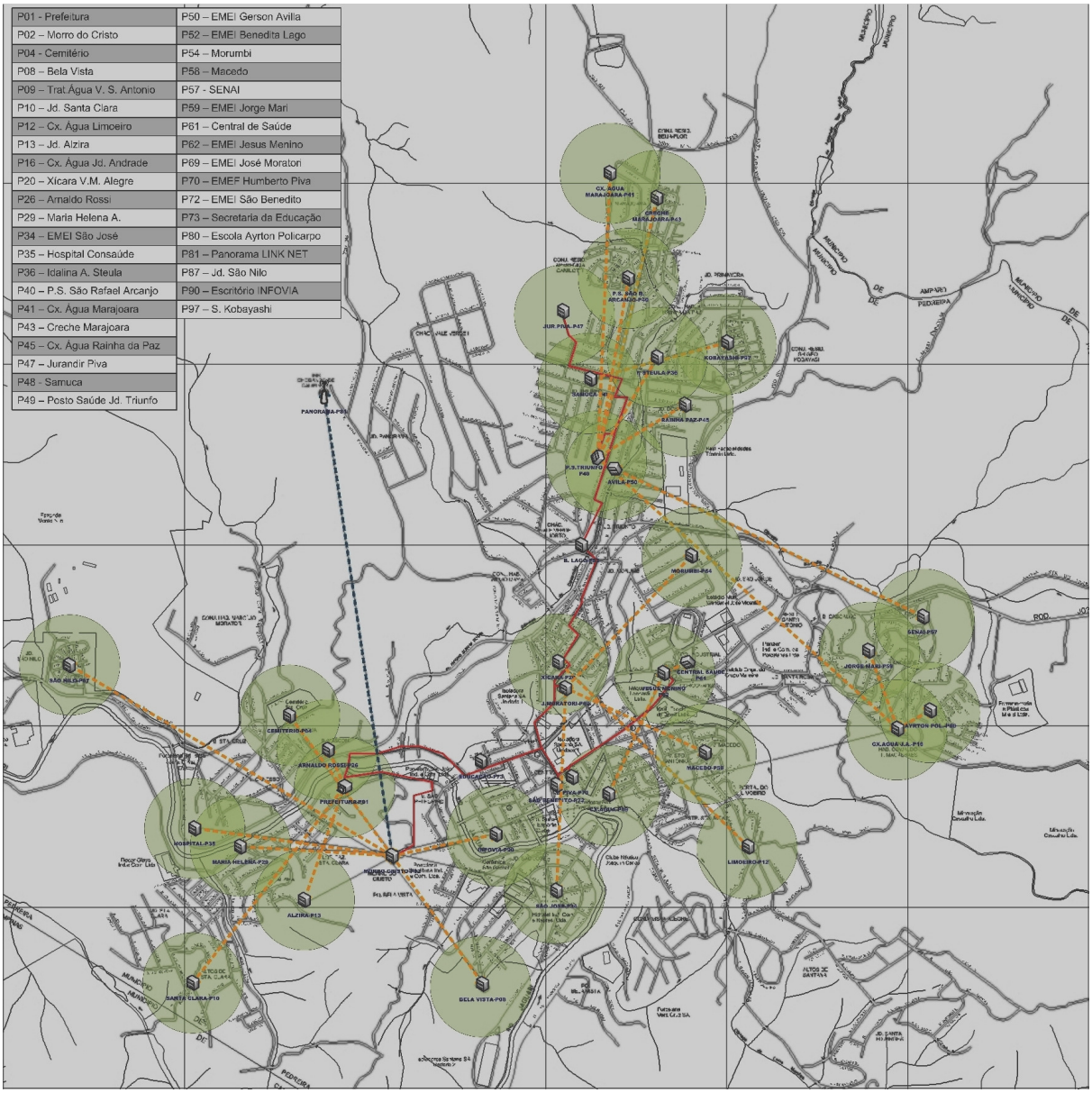

Figura 3 - Prédios Públicos - Pedreira

2.6 *Middleware*

De acordo com a IBM [24], um *middleware* é um software que conecta duas aplicações diferentes e transmite informações entre elas. Como exemplo, um *middleware* pode ligar um banco de dados a um servidor web.

O RNP[1][25] define *middleware* como um neologismo criado para designar camadas de software que não constituem diretamente aplicações, mas que facilitam o uso de ambientes ricos em tecnologias da informação. A camada de *middleware* concentra serviços como identificação, autenticação, autorização, diretórios, certificados digitais e outras ferramentas de segurança.

Stanton [26] complementa que um *middleware* deve possuir ferramentas para domar a complexidade de integração das aplicações criando uma segunda camada da infra-estrutura de TI, localizada acima da camada de rede. Trata-se da camada onde a tecnologia se encontra com as políticas e regras de negócio.

Aplicações tradicionais implementam todas as políticas e regras de negócio de forma independente, porém com a utilização de um *middleware* tem-se centralizada toda a lógica de interconexão entre os mais heterogêneos ambientes e aplicações de um sistema.

O intuito da criação de um *middleware* para cidades digitais é fornecer mecanismos de interconexão lógica entre os diversos serviços oferecidos pelos cidadãos em uma cidade digital. Este mecanismo deve ser capaz de encapsular toda complexidade da publicação de serviços na arquitetura de cidades digitais.

2.7 *Peer-to-peer*

Sistemas *peer-to-peer* (P2P) emergiram como um relevante fenômeno social e tecnológico nos últimos anos. Trata-se de sistemas distribuídos autônomos em que idealmente, todos os nós seguem as mesmas regras e dividem responsabilidades trocando informações e serviços entre os seus pares. Segundo Yang [27], a tecnologia P2P vem tornando-se uma popular alternativa para compartilhar grandes volumes de dados, pois os custos de armazenamento (espaço em disco, por exemplo) e largura de banda necessária para transferência destas informações é distribuída entre os *peers* (nós) da rede. Ripeanu[28], complementa que redes P2P podem ainda serem utilizadas para compartilhar ciclos de CPU, como por exemplo em projetos como o SETI@Home [29].

Em uma rede P2P, as responsabilidades são distribuídas entre os nós, pois todos atuam como cliente e servidor dos serviços. Cada nó compartilha com os seus pares recursos como

[1] Rede Nacional de Ensino e Pesquisa

largura de banda, espaço em disco e informações, criando desta forma uma rede descentralizada.

Segundo Sousa [30], a tecnologia *peer-to-peer*(P2P) tem a capacidade de disponibilizar recursos com maior robustez a um custo reduzido, maximizando a utilização dos recursos de cada dispositivo ligado à rede. Sousa ainda cita que a tecnologia P2P é uma extensão natural da filosofia da robustez através da descentralização da Internet. Outra vantagem da arquitetura P2P é que a infra-estrutura não depende de um servidor central e usualmente trabalha fora do sistema de nomeação da internet: o DNS. Podemos classificar como recurso qualquer dispositivo de hardware ou software de um elemento da rede (*peer*).

Nascimento [31] lista alguns benefícios da utilização de redes P2P:

1. Melhora da escabilidade e tolerância a falhas por meio da não existência de servidores centrais;
2. Redução do custo de infra-estrutura com o aproveitamento e utilização de recursos (como largura de banda, processamento e espaço em disco) de *peers* que são desconsiderados por outras abordagens como *Cloud Computing* [32];
3. Balanceamento de carga por meio da reutilização de recursos entre os *peers*;
4. Redundância e tolerância a falhas por meio da topologia descentralizada;

Redes *peer-to-peer* também tem sido amplamente utilizadas como uma eficiente forma de distribuição de VoD (*Video on demand*). Vídeo sob demanda é uma tecnologia desenvolvida com o intuito de possibilitar usuários selecionar e assistir vídeos através de uma rede de computadores, freqüentemente a Internet [33].

Usualmente as arquiteturas de VoD dependem de um ou somente poucos servidores de conteúdo para exercer a distribuição de vídeo. Esta topologia centralizada posiciona uma série de problemas nestes provedores, como exemplos a alta carga de serviço gerada pelo amplo número de clientes envolvidos nesta forma de distribuição de conteúdo e também a criação de um ponto único de falha, pois toda a estrutura depende de um único servidor central.

Muitas alternativas foram propostas na tentativa de minimizar este problema, como exemplos a utilização de servidores *proxys*, adoção de tecnologias como *multicasting* e broadcasting. Porém a alternativa que se apresenta mais viável a fim de solucionar os

problemas da distribuição de vídeos sob demanda é a utilização de tecnologias peer-to-peer (P2P)[34].

Como visto, os *peers* em uma comunicação P2P interagem entre si sem o uso de intermediários resolvendo muitos dos problemas inerentes a arquitetura tradicional cliente-servidor. Desta forma, os *peers* podem rapidamente lidar com um amplo e repentino aumento na demanda de vídeos, assim como aperfeiçoar a utilização da largura de rede. Especificamente nas aplicações VoD cada *peer* age como um cliente e um servidor de vídeo sob demanda, recuperando os trechos que ele está desprovido e compartilhando os trechos que já estão em seu domínio.

Já existem muitos exemplos na literatura da utilização de tecnologia P2P na distribuição de vídeo sob demanda como Chaining [33], DirectStream [35], P2Cast [34], P2VoD [34].

Segundo Damiani [36] podemos dividir as aplicações atuais de P2P em três categorias de acordo com o tipo de aplicação:

1. Compartilhamento de arquivos: Aplicações P2P como Napster, Gnutella, FreeNet, KaZaA e outras que possibilitam o compartilhamento de dados atavés da internet;
2. Processamento distribuído: Além de compartilhar arquivos, sistemas P2P de processamento distribuído compartilham poder computacional entre seus nós, como por exemplo, o projeto SETI@Home;
3. Mensagens instantâneas: programas como Msn Messenger, Yahoo! Messenger possibilitam aos usuários compartilhar mensagens de texto, mensagens de voz e vídeo e arquivos.

Já [37] e [38] classificam as redes *peer-to-peer* de acordo com a arquitetura de redes que podem ser:

- Centralizada: esta arquitetura possuí uma lista de *peers* constantemente atualizada hospedada em servidores centrais. Os nós utilizam estes servidores para encontrar os arquivos e serviços desejados. Como exemplo desta arquitetura podemos citar a rede Napster. Esta abordagem possui a desvantagem de apresentar-se como uma arquitetura não-escalável e o nó central representam um ponto único de falha (Figura 4).

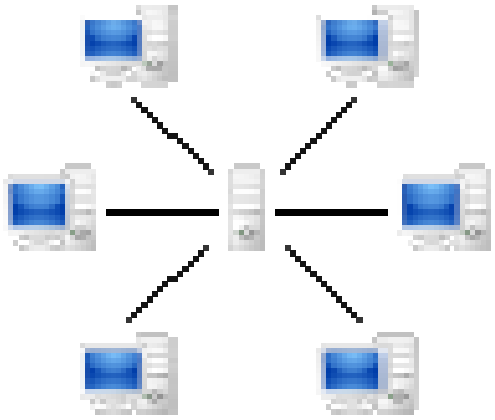

Figura 4 Arquitetura P2P Centralizada[68]

- Descentralizadas e desestruturadas: classificam-se nesta categoria sistemas em que não existem nenhum servidor central nem qualquer controle preciso sobre a topologia de rede, como exemplo a rede Gnutella. A rede é formada por *peers* (nós) que se conectam aleatoriamente a rede. A busca por um *peer* ou serviço da rede é realizada através de broadcasting aos nós vizinhos gerando uma grande sobrecarga de tráfego aos nós participantes da rede (Figura 5).

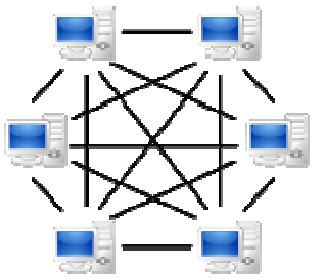

Figura 5 Arquitetura P2P Descentralizadas e desestruturadas

- Descentralizadas mas estruturadas: esta arquitetura não possui nenhum servidor central, mas possuem uma significativa estrutura. Por estrutura entende-se a

topologia P2P (as conexões entre os nós da rede) que é rigidamente controlada. Esta rigidez proporciona uma rápida localização dos nós e serviços da rede através de estruturas como, por exemplo, tabelas hash distribuídas (DHT). Como exemplo desta arquitetura podemos citar o JTXA. Este é o modelo arquitetural que será utilizada neste trabalho.

2.7.1 Jxta

A Tecnologia JXTA (do inglês *juxtapose*) é uma plataforma de rede e computação criada com a finalidade de resolver uma série de problemas de computação distribuída, especialmente na área amplamente referida como *peer-to-peer*, ou simplesmente P2P[39]. Esta tecnologia foi originalmente concebida pela Sun Microsystems[40] e teve seu desenvolvimento amparado por diversas instituições acadêmicas e a indústria.

Anteriormente a esta plataforma, os sistemas P2P eram desenvolvidos isoladamente, implementando apenas poucos serviços e se baseando em protocolos incompatíveis com outros sistemas. Cada sistema cria sua própria comunidade P2P, duplicando os esforços na criação de uma infra-estrutura para um sistema P2P. Com o objetivo de criar uma compatibilidade e encapsular tarefas comuns na criação de um sistema P2P é desenvolvida a tecnologia JXTA.

De acordo com [39] o projeto JXTA possui uma série de objetivos que são derivados das necessidades dos sistemas P2P existentes e os que ainda estão em desenvolvimento, entre estes podemos destacar:

- Interoperabilidade entre variados sistemas P2P;
- Independência de plataforma de desenvolvimento;
- Operar sob as mais variadas tecnologias (sensores, PDAs, roteadores, computadores, etc.).

Traversat [41] acrescenta que o JXTA foi projetado com a visão de um mundo onde cada *peer*, independente de plataforma de software ou hardware possa ser beneficiado do fato de estar conectado a milhões de outros *peers*.

O projeto JXTA define uma camada virtual genérica de rede que pode ser utilizada pelas mais variadas aplicações P2P. Esta camada virtual possibilita que cada *peer* se comunique com qualquer outro *peer* independente da sua localização de rede. As mensagens são

roteadas de forma transparente, mesmo através de firewalls ou NATs, e podem utilizar diferentes protocolos de transporte como TCP/IP ou de aplicação como HTTP para localizar os *peers*. [42]

Traversat [41] afirma que a tecnologia JXTA encapsula o entendimento e a complexidade da comunicação entre os *peers* dos mais heterogêneos sistemas P2P e tecnologias de rede utilizadas. Ela padroniza a maneira que os *peers* encontram outros *peers*, organizam-se em grupos, trocam mensagens e se monitoram.

A arquitetura JXTA é constituída de três camadas distintas de acordo com a Figura 6.

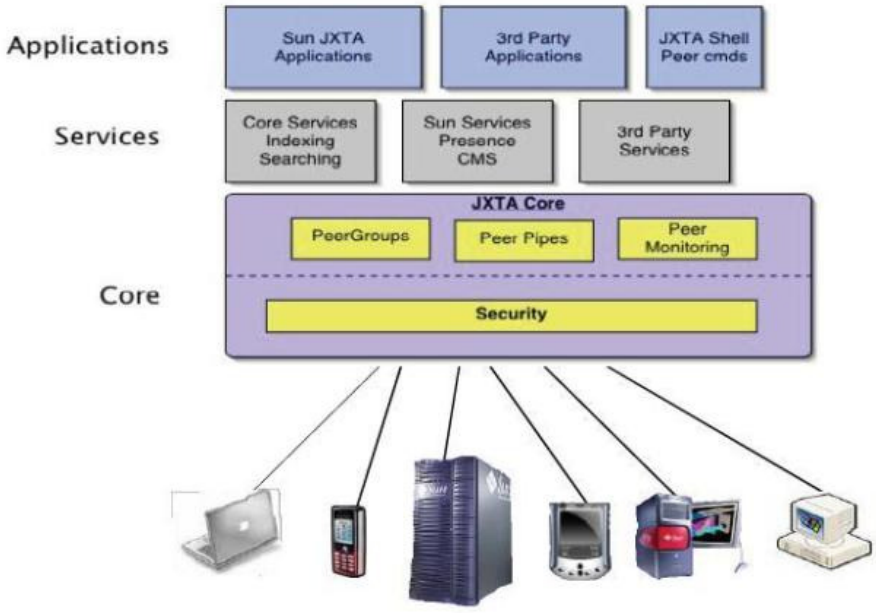

Figura 6 - Camadas JXTA[68]

Na Figura 6 percebemos a camada principal “Core”, especifica protocolos que definem os mecanismos de comunicação nas mais diversas plataformas (computadores, servidores, impressoras), protocolos(DNS, NAT, Firewall, DHCP), linguagem (JAVA, C, C++), mecanismo de associação a grupos (*peer groups*), mecanismos de monitoração e segurança.

A camada Services provê funcionalidades, utilizando os protocolos da camada JXTA Core, que podem ser utilizadas pelas mais diversas aplicações. Ela especifica um conjunto de interfaces para que qualquer desenvolvedor crie seu serviço específico [31]. A camada Applications JXTA é a camada onde as aplicações que utilizam recursos do JXTA são inseridas, como por exemplo, o *middleware* proposto por este trabalho.

A arquitetura JXTA[43] pode ser classificada como uma topologia descentralizada mas estruturada. Neste modelo é criada uma rede de servidores "*Rendezvous*·" de forma a estabelecer interconexões entres os *peers*. Desta forma é gerado um grafo completo onde cada servidor desempenha o papel de um vértice e as conexões entre estes servidores se apresentam como arestas.

2.8 *Web Semântica*

Segundo o W3C [44] a Web Semântica (WS) fornece uma estrutura comum que permite que dados sejam compartilhados e reutilizados em todas as aplicações web. Em 2001, Berness-Lee [45] definiu web semântica como uma extensão da internet atual mas com uma infra-estrutura que possibilita a compreensão e o gerenciamento dos conteúdos armazenados na web.

A visão da Web Semântica é estender os princípios da Web tradicional alterando o enfoque principal de documentos para dados. Os dados devem ser acessados usando a arquitetura da Web em geral, utilizando, por exemplo, a URI[2]. Os dados também devem ser relacionados uns aos outros assim como os documentos (ou partes de documentos) já estão relacionados na web tradicional. Isto também significa a criação de um framework padrão que permita que dados sejam compartilhados e reutilizados nas mais diversas áreas e aplicações para serem processados automaticamente por ferramentas baseadas neste padrão. [45]

Carvalho [32] também define web semântica como uma extensão da web atual, na qual é dado um significado bem definido para a informação, permitindo que computadores e pessoas trabalhem em cooperação.

Basicamente a web semântica é composta por três elementos [46]:

a) Representação do conhecimento: a WS proporciona a inserção de conteúdo significativo (aquele que possui valor semântico) a páginas web criando desta

[2] URI - Uniform Resource Identifier é uma cadeia de caracteres compacta usada para identificar ou denominar um recurso na Internet.

forma mecanismos para que os agentes interpretem e classifiquem o conteúdo de uma página;

b) Ontologias: ontologia é a especificação de uma conceituação. Tratando especificamente de WS, uma ontologia cria ligações terminológicas entre membros de uma comunidade. Ontologia também pode ser classificada como um arquivo que define a relação entre grupo de termos. Por exemplo, o termo "irmão" se refere a uma relação familiar entre pessoas que compartilham o mesmo conjunto de pais. Segundo Berners-Lee [45] o uso de ontologias eficazes e compreensivas são peça chave para a criação da WS. Estas são representadas na forma de metadados. Metadados, são dados que contém informação a respeito de atributos de um recurso. Geralmente um metadado é uma informação inteligível por um computador incluída em páginas Web. Os metadados facilitam o entendimento dos relacionamentos e a utilidade das informações compostas pelos dados. Oliveira [46] define um metadado como "um dado associado a algum objeto que auxilia seus usuários potenciais a obter vantagem completa do conhecimento da sua existência ou características". Como exemplo de linguagens de metadados podemos citar o XML[3].

c) Agentes: resumidamente, um agente de WS é um software que percorre a Internet buscando informações relevantes. Estas informações são obtidas através das mais variadas fontes, processadas, expressadas e entendidas através das regras especificadas pelas ontologias.

Em [47] podemos encontrar diversos estudos de caso a respeito de aplicações na área de WS, entre eles podemos destacar a cidade de Zaragoza (Espanha) e o Principado de Astúrias.

A cidade de Zaragoza aplicou Web Semântica para facilitar o acesso dos cidadãos aos serviços disponibilizados pela prefeitura. Usualmente a linguagem utilizada pelos cidadãos difere da utilizada pela administração pública. Ruiz [48] cita as diferenças de linguagem utilizadas pelo cidadão que deseja buscar um serviço onde possa descartar um eletrodoméstico antigo ao serviço de recolhimento oferecido pela prefeitura (Recolhimento de

[3] XML (*eXtensible Markup Language*) é uma recomendação da W3C para gerar linguagens de marcação para as mais diversas necessidades.

grandes itens). Ao realizar uma busca no portal por “descartar máquina de lavar” o cidadão não é encaminhado ao serviço específico, pois o portal não encontra a palavra chave que encaminha a solicitação a este recurso.

Para solucionar este problema Ruiz [48] desenvolveu uma máquina de busca apresentada na Figura 7 capaz de localizar serviços online baseados no significado tanto da busca realizada pelo usuário quanto pela descrição do serviço.

Figura 7 - Portal de Serviços de Zaragoza [48]

Alguns dos benefícios da utilização de WS pela cidade de Zaragoza incluem [48]:

1. Proporcionar um melhor serviço aos seus cidadãos e, assim, melhorar a imagem da administração pública;
2. Estimular o *e-Government*, proporcionando um acesso mais fácil à informação relevante a custo reduzido, diminuindo a carga sobre o *call center* e escritórios físicos;
3. Fácil interação, seja através de palavras-chave ou de linguagem natural de alta precisão;
4. Capacidade de sugerir serviços complexos através de um motor de busca que "compreende" o cidadão;
5. Habilidade de sugerir serviços co-relatos.

O principado de Astúrias [49] também aplica WS para aprimorar motores de busca em seu portal. Especificamente foi criada uma semântica capaz de aproximar os documentos de administração pública (escritos utilizando jargões administrativos) da linguagem do cidadão.

O usuário pode expressar a suas buscas utilizando as suas próprias palavras e o portal (através das ontologias) analisa a expressão e identifica a semântica da solicitação. A Figura 8 demonstra uma busca por "*dinero para Iglesias*" (dinheiro para igrejas) e interpreta a solicitação identificando todos os documentos com relações monetárias e igrejas.

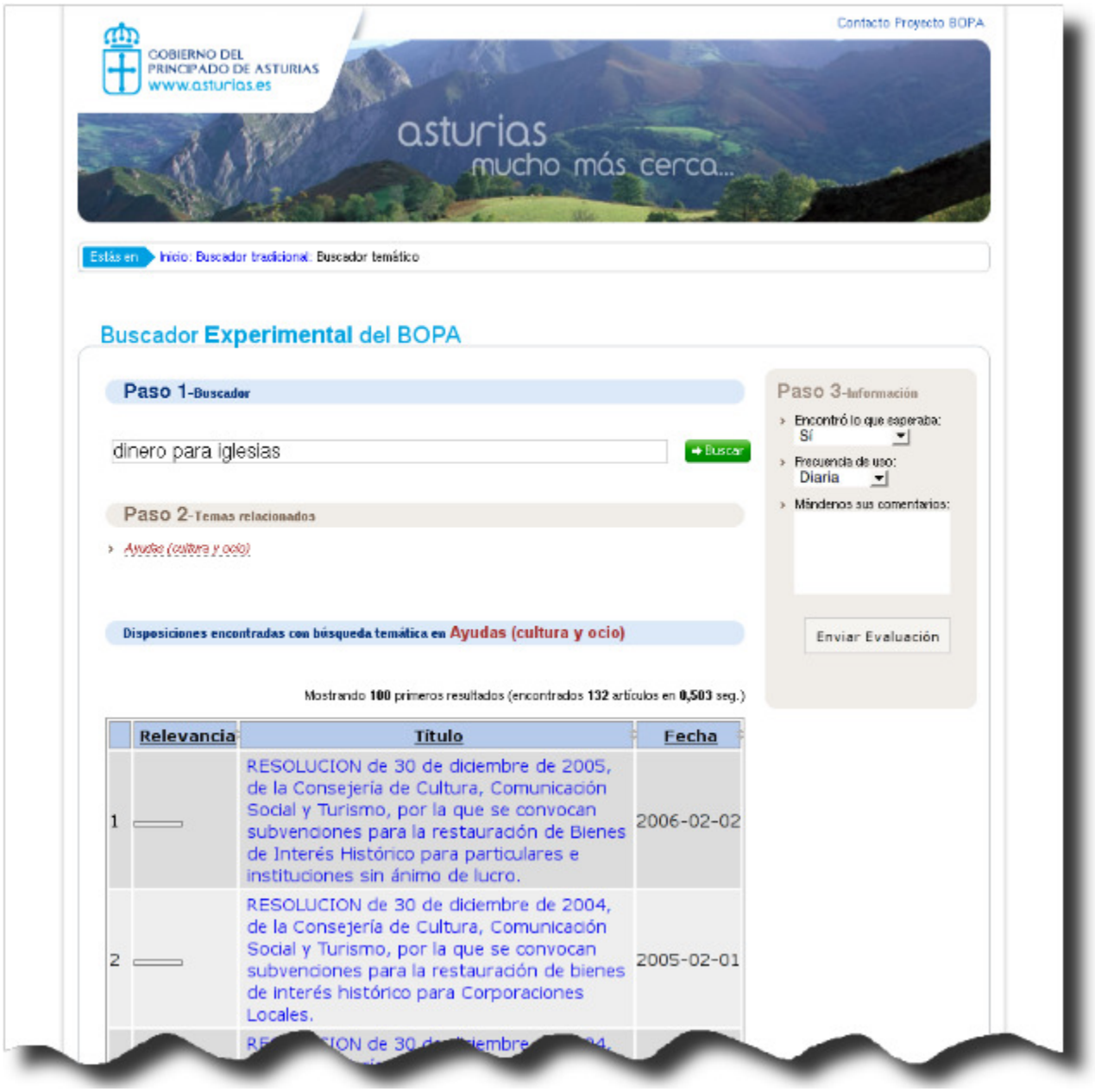

Figura 8 - Portal de Asturias

Berrueta[46] lista alguma das vantagens obtidas com a utilização da WS:

- Usuários podem utilizar suas próprias palavras e ainda obter resultados, mesmo se não houver correspondência direta entre os termos de busca e os termos do documento;
- Usuários podem descobrir os novos documentos que estão relacionados à sua busca inicial;
- Os usuários podem obter sugestões sobre como melhorar suas consultas (por exemplo com a utilização de conceitos semanticamente relacionados).

A utilização da Web Semântica possibilitou ao Principado de Asturias [49] o desenvolvimento de uma ferramenta de pesquisa mais poderosa, sem um aumento significativo da complexidade da interface apresentada ao usuário.

Desta forma, verifica-se a necessidade da utilização da web semântica para construção dos mecanismos de busca do portal relacional comunitário, desenvolvido neste trabalho, de forma a melhorar a qualidade dos serviços prestados aos cidadãos. Esta melhora justifica-se através da utilização de uma linguagem natural na busca pelos serviços estimulando a rápida localização e consumo dos serviços da Cidade Digital.

3 Arquitetura de Cidades Digitais

Neste capítulo será detalhada a arquitetura para cidades digitais inicialmente proposta por Panhan, Ignatowicz, Santos e Mendes [12] [10] e ampliada neste trabalho. Esta arquitetura visa criar ambientes de interoperabilidade digital semelhante às relações no mundo real, proporcionando assim a facilidade na interação de serviço entre os cidadãos. Também apresentaremos o *middleware* proposto para esta arquitetura. Este *middleware* é projetado de forma a fornecer mecanismos de interconexão lógica entre os diversos serviços oferecidos pelos cidadãos em uma cidade digital.

Em [50] é definido um conjunto mínimo de premissas, políticas e especificações técnicas que regulamentam a utilização da Tecnologia de Informação e Comunicação (TIC) pelos órgãos governamentais, estabelecendo as condições de interação entre a iniciativa pública. Partindo deste conjunto básico de premissas nesta seção será apresentada uma arquitetura baseada em um m*iddleware* estruturado na Plataforma de Redes P2P, estendendo os modelos tradicionais de interoperabilidade nos governos para outros segmentos da sociedade como cidadãos, empresas e o terceiro setor [10] (Figura 9).

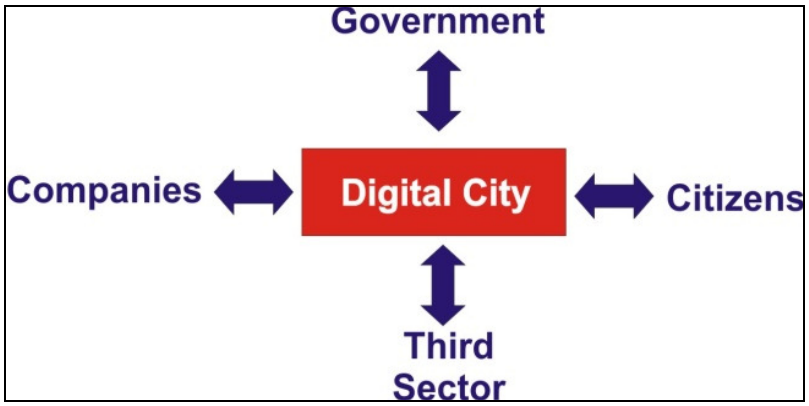

Figura 9 - Segmentos de uma cidade digital

Para seguir as premissas estabelecidas por Yamaoka [50], a arquitetura proposta deve ser divida em quatro grandes áreas cada qual responsável pela interoperabilidade dos diversos segmentos da sociedade:

- Interconexão;
- Segurança;
- Meios de Acesso;
- Organização e Intercâmbio de Informações.

Esta divisão nos permite dividir responsabilidades e criar camadas tão discretas e independentes quanto possível. Para cada uma dessas áreas foram especificados componentes, os quais são estabelecidos padrões. Componentes são módulos da arquitetura projetados para cumprir uma determinada função. Os padrões são estabelecidos para especificar o desenvolvimento destes componentes. As próximas seções detalharão os componentes e padrões definidos para cada área do modelo proposto.

3.1.1 Interconexão

O segmento “Interconexão” estabelece regras para interconexão entre os mais variados segmentos da sociedade em uma rede de dados. Basicamente podemos dividir estes segmentos em iniciativa pública, iniciativa privada (empresas prestadoras de serviço e b2b) e terceiro setor. Estes podem tanto prover como consumir serviços disponibilizados na Cidade Digital. Esta rede pode tratar-se de uma MAN (Metropolitan Area Network), uma Open MAN [5] ou até mesmo a Internet.

Para o modelo proposto, a implementação da interconexão de Cidades Digitais é realizada sobre uma Open MAN, pois esta solução é capaz de prover livre acesso a informação e ao conhecimento. Além disto, de acordo com Mendes [5], esta solução oferece um ambiente único genérico e pode ser considerado o melhor candidato para tornar-se a solução de escolha para uma única rede convergente e universal.

3.1.2 Segurança

Este segmento trata dos aspectos de segurança de TIC que a arquitetura proposta para a construção das Cidades Digitais deve considerar. São tratados padrões como Segurança IP, Criptografia, Desenvolvimento de Sistemas e Serviços de Rede.

A implementação de políticas de segurança de informação em uma cidade digital deve ser tratada com responsabilidade, pois é de vital importância para a redução de riscos a todos os setores da sociedade. Esta definição pode seguir programas e políticas estabelecidas no mercado como, por exemplo, ITIL, ISO e COBIT.

ITIL (*Information Technology Infrastructure Library*) é um modelo de referência criado para o gerenciamento de serviços ou processos de TI. Este modelo descreve processos necessários para a gerência eficaz de uma infra-estrutura de TI de modo a garantir níveis de serviços adequados a uma organização.

Já a ISO é uma compilação de recomendações para melhores práticas de segurança, que podem ser aplicadas por empresas, independente do seu porte ou setor [51]. Cobit (*Control Objectives for Information Technology*) é um guia de gestão de TI que inclui recursos tais como técnicas de gerenciamento, auditoria, ferramentas de implementação, entre outros. A metodologia pode ser divida em planejamento, organização, aquisição, implementação, entrega, suporte e monitoração.

Yamaoka [50] estabelece diretrizes em relação à segurança de uma cidade digital. Estas diretrizes podem ser classificadas em três categorias:

1. Requisitos
    a. Os requisitos de segurança da informação, dos serviços e de infra-estrutura devem ser identificados e tratados de acordo com a classificação da informação, níveis de serviço definidos e resultado da análise de riscos.
    b. Os dados, informações e sistemas de informação do governo devem ser protegidos contra ameaças de forma a reduzir riscos e garantir a integridade, confidencialidade e disponibilidade.
    c. A segurança deve ser tratada de forma preventiva. Para os sistemas que apóiam processos críticos devem ser elaborados planos de continuidade, nos quais serão tratados os riscos residuais visando atender os níveis mínimos de produção.

d. A segurança é um processo que deve estar inserido em todas as etapas do ciclo de desenvolvimento de um sistema.
e. A documentação dos sistemas, dos controles de segurança e das topologias dos ambientes deve ser mantida atualizada e protegida.
f. Os usuários devem conhecer suas responsabilidades com relação à segurança e devem estar capacitados para a realização de suas tarefas e utilização correta dos meios de acesso

2. Padrões
    a. Os serviços de segurança de XML devem estar em conformidade com as especificações do W3C[52].
    b. O uso de criptografia e certificação digital, para a proteção do tráfego, armazenamento de dados, controle de acesso, assinatura digital e assinatura de código, deve estar em conformidade com as regras da ICP-Brasil [40].
3. Soluções
    a. As informações sensíveis que trafegam em redes inseguras, incluindo as sem fio, devem ser cifradas, de modo adequado.
    b. Os dados e informações devem ser mantidos com o mesmo nível de proteção, independente do meio em que estejam sendo processados, armazenados ou trafegando.
    c. Os sistemas devem possuir registros históricos (*logs*) para permitir auditorias e provas forenses, sendo imprescindível a adoção de um sistema de sincronismo de tempo centralizado, bem como se devem utilizar mecanismos que garantam a autenticidade dos registros armazenados, se possível com assinatura digital.
    d. Nas redes sem fio metropolitanas recomendam-se a adoção de valores randômicos nas associações de segurança, diferentes identificadores para cada serviço e a limitação do tempo de vida das chaves de autorização

Sendo assim, o modelo proposto sugere, na medida do possível, a adoção dos modelos propostos pelo ITIL, Cobit e ISO em conjunto da implementação das soluções, requisitos e padrões sugeridos por Yamaoka [50] para cidades digitais.

3.1.3 Meios de acesso

No segmento Meios de Acesso são explicitadas as questões relativas aos padrões dos dispositivos de acesso aos serviços oferecidos pelas Cidades Digitais. Podendo definir como serão tratados dispositivos como, por exemplo, estações de trabalho, telefone celular, laptops, televisão digital, balcão de serviços, etc.

Os sistemas desenvolvidos devem ser projetados de maneira a fornecer recursos de acessibilidade aos cidadãos portadores de necessidades especiais, a grupos étnicos minoritários e aqueles sob risco de exclusão social ou digital. O atendimento via balcão de prestação de serviços deve ser considerado em toda a sua abrangência, de forma a possibilitar que os benefícios decorrentes do uso dos serviços de governo eletrônico venham a ser estendidos à camada da população que não pode ter acesso direto a esses serviços por meio dos dispositivos previstos [50].

Estes sistemas quando utilizarem a Internet como meio de comunicação e estações de trabalho como dispositivo de acesso, estes devem ser projetados para fornecer acesso através de navegadores web (*browsers*).

3.1.4 Organização

Este segmento lida com aspectos relacionados ao processamento e intercâmbio de informação através dos serviços oferecidos pelas Cidades Digitais. Estes incluem componentes para troca de informações, processamento e definição de dados e catálogo de Padrões de Dados (CPD). Para a execução destes serviços e o intercâmbio de informações faz-se necessário a construção de estruturas rígidas de metadados (Xml) amparados pelos conceitos de Web Semântica. A definição e implantação dos padrões deste segmento é um dos pontos chaves deste trabalho. Desta forma, o capítulo 4 explicitará os padrões definidos para esta arquitetura.

Para conquistar interoperabilidade, todos os segmentos da sociedade devem estar engajados num esforço contínuo para assegurar que sistemas, processos e culturas de uma cidade real sejam gerenciados e direcionados para maximizar oportunidades de troca e

reuso de informações. Com este intuito, faz-se necessária uma definição clara de políticas e especificações (definidas no capitulo 4) para interoperabilidade e gerenciamento de informação entre cidadãos, empresas, governo e provedores de serviço.

3.1.5 Benefícios da utilização da arquitetura proposta

A Cidade Digital é projetada como uma estrutura básica para o planejamento estratégico tanto da iniciativa pública quanto privada em diversos cenários de negócios (B2B, B2C, B2G, G2G, G2C e C2C). Esta abordagem permite a racionalização de investimentos em T.I., através do compartilhamento, resumo e intercâmbio de recursos tecnológicos.

Os recursos de informação estratégica das empresas, governo e prestadores de serviços constituem valiosos ativos econômicos. Ao garantir que a informação ou serviço possa ser rapidamente localizada e transmitida entre os setores públicos e privado, resguardadas as devidas obrigações de privacidade e segurança, cria-se um grande impulso na economia local.

Atualmente, inúmeras nações do mundo identificam a necessidade de altos investimentos no desenvolvimento de acesso universal dos seus cidadãos a redes multi-serviço. Segundo Mendes [5] o amplo acesso a estes serviços provoca um aumento imediato do PIB (produto interno bruto). Logo, pode-se notar que o acesso a estes serviços, particularmente quando lidamos com sistemas de e-Gov, pode ser alcançado através da criação de uma cidade virtual (ou Cidade Digital). A criação de Cidades Digitais é essencialmente uma questão de estabelecer políticas de desenvolvimento, procedimentos e normas para as TCI[4] [10]. Estas visam criar estruturas dedicadas com o propósito de alcançar locais vitais à interoperabilidade buscando melhor qualidade na prestação de serviços a custos reduzidos.

Uma das formas de reduzir os custos destas estruturas é com a criação de um *middleware*[12] como base para a criação de serviços eletrônicos. Segundo Panhan[12] a criação de um *middleware* é um pré-requisito para uma melhor prestação de serviços a um custo reduzido para todos os membros da Cidade Digital. Uma sociedade moderna e integrada exige sistemas igualmente modernos e integrados, interoperáveis, trabalhando de

[4] TIC - Tecnologias de Informação e Comunicação

forma íntegra, segura e coerente. Este *middleware*, cuja arquitetura é apresentada na Figura 10, é uma solução para esta exigência.

Na Figura 10, podemos visualizar os possíveis públicos alvos da cidade digital (cidadãos, empresas, governo etc.) conectados por meio de variados dispositivos de acesso ao portal da Cidade Digital. Uma vez conectado a este portal, ele tem acesso aos diversos serviços oferecidos pela Cidade Digital.

Os prestadores de serviço oferecem seus produtos a Cidade Digital através do *middleware*. Uma vez conectado ao *middleware* o serviço é apresentado ao usuário na forma de uma interface comum que é o portal relacional comunitário.

Desta forma a arquitetura proposta para Cidades Digitais abrange a troca de informações pelos mais diversos segmentos da sociedade.

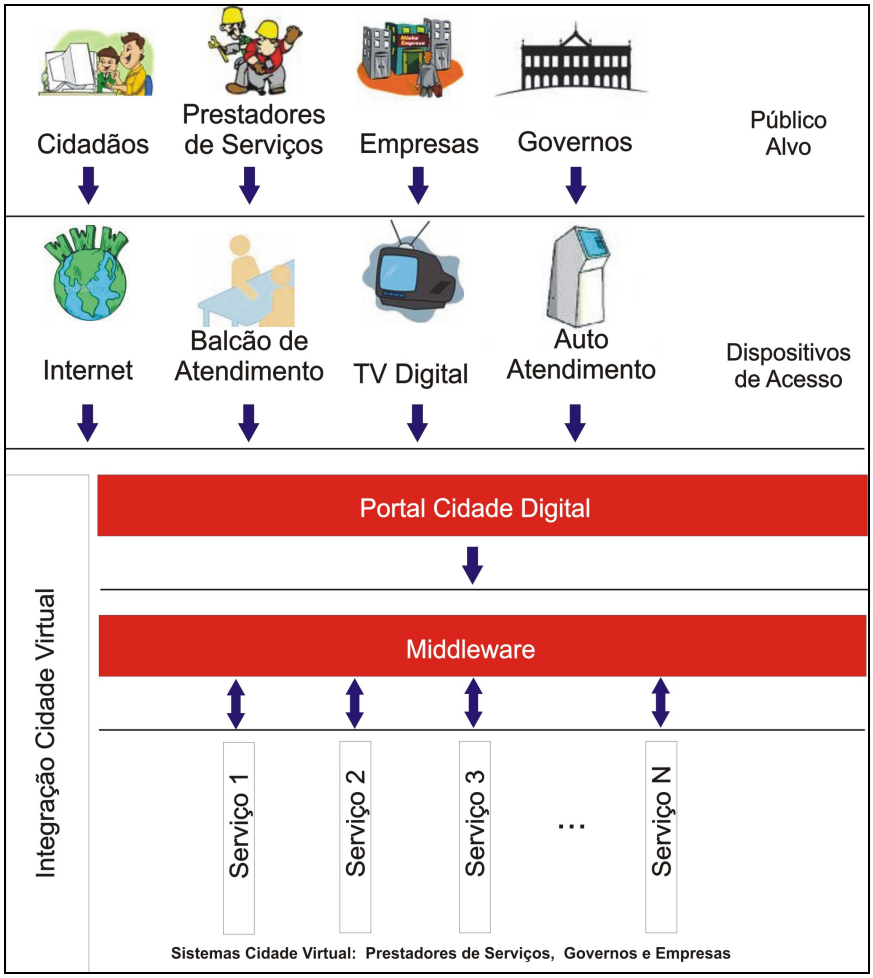

Figura 10 - Arquitetura para cidades digitais

3.1.6 Middleware

A arquitetura proposta por este trabalho de uma cidade digital é construída sobre uma rede *peer-to-peer* devido em grande parte a capacidade de cada nó da rede fornecer serviço a outros computadores. Como visto no Capítulo 2, outra vantagem da arquitetura *peer-to-peer* é que a infra-estrutura não depende de um servidor central e usualmente trabalha fora do sistema de nomeação da internet: o DNS[5]. Esta é uma vantagem crucial em relação à *cloud-computing*, pois esta tem certa dificuldade em atingir máquinas isoladas por um sistema de NAT[6] ou firewall. Em uma rede P2P, as responsabilidades são distribuídas entre os nós, pois todos atuam como cliente e servidor dos serviços. Cada nó compartilha com os seus pares recursos como largura de banda, espaço em disco, informações criando desta forma uma rede descentralizada.

Tratando-se especificamente da arquitetura de uma Open MAN[5], uma rede *peer-to-peer* apresenta uma importante vantagem em relação as demais abordagens quando lida com distribuição de conteúdo como por exemplo *Video on demand*. Como visto em [5], internamente a uma Open MAN a largura de banda é praticamente infinita e de baixo custo. Após um nó da rede requisitar um dado externo a Open MAN, ele próprio atuará como servidor local dos dados adquiridos a outros *peers* da rede. Desta forma a requisição de uma informação externa a rede é feita apenas uma vez, economizando recursos e aproveitando-se da alta capacidade de transmissão de dados na Open MAN. Os *peers* interagem entre si sem o uso de intermediários resolvendo muitos dos problemas inerentes a arquitetura tradicional cliente-servidor.

Podemos encontrar muitos exemplos na literatura da utilização de tecnologia P2P na distribuição de vídeo sob demanda como Chaining [33], DirectStream [35], P2Cast [34], P2VoD [34].

Segundo Sousa [30], A tecnologia P2P tem a capacidade de disponibilizar recursos com maior robustez a um custo reduzido maximizando os recursos de cada dispositivo ligado à rede. Sousa ainda cita que a tecnologia P2P é uma extensão natural da filosofia da robustez através da descentralização da Internet.

---

[5] DNS (*Domain Name System* - Sistema de Nomes de Domínios)

[6] NAT (*Network Address Translation*) é a mudança de um endereço de IP usado em uma rede para outro diferente conhecido dentro da outra rede.

No Capítulo 2 foi apresentado três diferentes arquiteturas de redes P2P. A arquitetura P2P utilizada neste trabalho é a arquitetura “Descentralizada mas estruturada”, pois esta possui as vantagens de possuir uma relativa estrutura mesmo não possuindo um servidor central (ponto único de falha). Esta rigidez proporciona uma rápida localização dos nós e serviços da rede através de estruturas como, por exemplo, tabelas *hash* distribuídas (DHT). Como exemplo desta arquitetura podemos citar o JTXA.

Um *middleware* é construído de forma a encapsular toda complexidade da publicação de serviços na arquitetura de cidades digitais fornecendo mecanismos de interconexão lógica entre os diversos serviços oferecidos pelos cidadãos em uma cidade digital.

Para a construção do *middleware*, é realizada a adição de uma camada de abstração na arquitetura P2P JXTA [43]. Esta camada extra atuará como controlador de acesso às redes das cidades digitais existentes.

Como visto no capítulo 2, a arquitetura *Juxtapose* (JXTA) é uma especificação independente de linguagem e plataforma para comunicação de dispositivos em redes *peer-to-peer* desconsiderando sua localização física e tecnologia de rede no qual se encontram instalados. Ela foi inicialmente proposta pela Sun Microsystems em 2001 [43] e é uma plataforma livre e tem como padrão a criação de um modelo de rede virtual sobre um modelo físico já existente, usualmente TCP/IP. Devido a esta característica possui a capacidade de atingir nós dispostos em um ambiente de NAT ou Firewall (sub redes TCP/IP, por exemplo). Outra característica é a possibilidade de criação de grupos de trabalho e gerência desta forma facilitando a administração da rede. Estas duas características combinadas justificam a adoção do JXTA para arquitetura do *middleware* proposto por Panhan[15] e [12].

Esta especificação foi desenvolvida na forma de uma arquitetura *microkernel*, ou seja, mesmo os serviços mais básicos estão implementados como módulos, restando para o kernel propriamente dito poucas funções, em geral, de comunicação entre os próprios módulos. Portanto, serviços básicos como a descoberta de nós (*peers*) ou de recursos na rede, comunicação entre dois ou mais nós, entre outros, são todos providos por módulos específicos.

Embora a descrição deste *middleware* não seja o enfoque deste trabalho, as próximas subseções serão dedicadas a tratar brevemente de temas que refletem a parte estrutural da arquitetura e o modelo de organização e acesso aos serviços disponibilizados na rede. A

definição completa do *middleware* proposto para a arquitetura pode ser encontrada em [15] e [12].

3.1.6.1 Estrutura

A estrutura básica do *middleware* toma como fundamento a topologia de rede overlay JXTA [16]. Este modelo cria uma rede de servidores "*Rendezvous*[7]" de forma a estabelecer interconexões entres estes. Desta forma é gerado um grafo completo onde cada servidor desempenha o papel de um vértice e as conexões entre estes servidores se apresentam como arestas.

Na arquitetura JXTA cada servidor mantém uma lista atualizada de *peers* conectados a rede. Deste modo é possível que uma busca por um serviço presente na rede seja feita em ordem linear quando a rede de servidores estiver consistente. Quando detectada uma inconsistência da rede outro mecanismo de busca é acionado (Randon Walker[53]). Este mecanismo retorna o localizador de um determinado *peer* em um tempo *n*, sendo *n* o número de servidores.

A Figura 11, representa o *middleware* implementado por Panhan[12]. Nesta arquitetura os *peers* Rx provêem roteamento, os Ex serviços e o Px reserva-se a fornecer acesso do portal de interação da cidade digital com a rede da cidade digital.

[7] **Rendezvous Peers** são nós especiais na topologia da rede que atuam como pontos de encontro para nós que não conseguem utilizar mecanismos de descoberta automáticos.

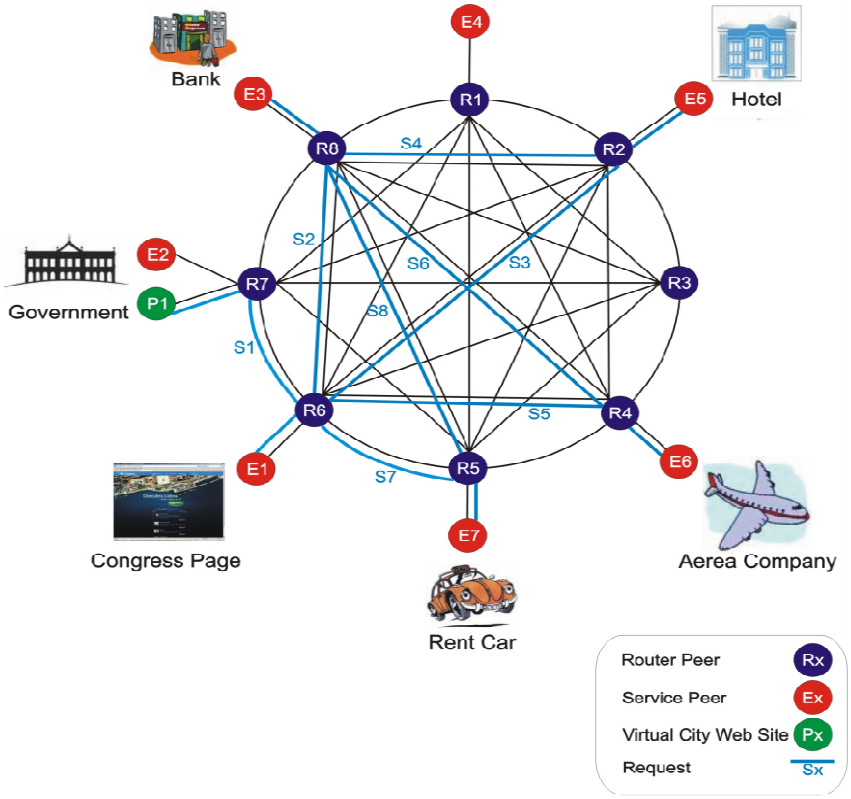

Figura 11 - Middleware de uma cidade digital [12]

3.1.6.2 Segurança

Uma preocupação quando se trata da criação de um ambiente corporativo ou governamental é quanto à segurança de dados, seja no armazenamento quanto no transporte.

Para prover a segurança necessária nas aplicações e nas transações de dados, podemos utilizar recursos providos pela API JXTA [53] [54], que fornece uma base para o controle de segurança nas aplicações desenvolvidas através de um conjunto de algoritmos específicos.

Dentre os atributos de segurança fornecido pela API JXTA destacam-se:

- TLS (Secure Transport Layer) – que cria um canal de segurança baseado na tecnologia de chave pública,
- Peer certificate – fornece a certificação dos peers,
- Personal Security Environment – em que todo peer é protegido por um peer ID e uma senha.

3.1.6.3 Catálogo

Para manter um conjunto consistente de identificadores de serviços na rede interna das cidades digitais, podem ser criados agentes atrelados ao portal das cidades digitais. Estes têm por finalidade realizar pesquisas temporárias na rede P2P para verificar os serviços ativos e relacionar a descrição do serviço com o identificador do mesmo.

Tendo em mãos este mapeamento, a criação, ordenação e disponibilização desses dados ficam a cargo de cada cidade por meio do portal da cidade digital. O próximo capítulo descreverá detalhadamente a definição destes requisitos.

4 Portal Relacional Comunitário

Neste capítulo, será detalhado o portal relacional comunitário construído para validar a proposta deste trabalho. A próxima seção descreverá a motivação para o desenvolvimento deste portal. A seção 4.2, detalha a arquitetura proposta. Em seguida são detalhados os padrões organizacionais desenvolvidos para a Cidade digital.

4.1 Motivação

O objetivo do Portal Relacional Comunitário é prover uma interface amigável para que cidadãos acessem os serviços disponibilizados pela Cidade Digital.

Este portal também visa permitir uma simples disponibilização de serviços a comunidade. Estes serviços serão criados tanto pela iniciativa pública quanto privada através de um sistema de comunicação web baseado no *middleware* proposto por Panhan[12].

Entre os serviços que podem ser disponibilizados pela iniciativa pública podemos citar: Pesquisa de débito imobiliário, Certidão negativa de mobiliário, Validação de certidão mobiliária e ISS, Pesquisa de débito, Certidão negativa de débitos imobiliários, Validação de certidão e Certidão de valor venal, Consulta o acervo da biblioteca municipal, etc.

Já a iniciativa privada pode negociar produtos e serviços e exibir um catálogo com as atividades desenvolvidas pela empresa dentre outras atividades. Para efetuar a disponibilização de um serviço, a iniciativa privada deve seguir os padrões de comunicação estabelecidos e realizar um cadastro no gestor do portal. Este gestor geralmente será a própria prefeitura ou uma ONG local. Este cadastro visa verificar a idoneidade dos provedores de serviço evitando assim possíveis tentativas de fraude na oferta de serviços ao cidadão.

O portal é projetado como uma estrutura básica para os mais diversos cenários de negócios (B2B, B2C, B2G, G2G, G2C e C2C).

Por tratar-se de um portal relacional comunitário disponibilizado pela Cidade Digital, este possui custos reduzidos de utilização permitindo assim a racionalização dos investimentos

em T.I. . Ao garantir que a informação ou serviço possa ser rapidamente localizada e transmitida entre os setores públicos e privado, cria-se um grande impulso na economia local.

4.2 Cenário de Exemplo

Esta seção exemplificará em linhas gerais como é realizada a oferta e consumo de um serviço através do portal relacional comunitário.

O portal pode ser visualizado de duas maneiras distintas: Dos cidadãos que consomem serviços oferecidos pelo portal e dos usuários (iniciativa pública e privada) que fornecem serviços a comunidade.

Para conectar à Cidade Digital, o cidadão necessita registrar-se no portal. Este registro fornece a identidade digital do usuário. Somente com estas credenciais, ele poderá acessar os serviços oferecidos pelo portal.

Um serviço é disponibilizado no portal através de um provedor. O registro de um provedor é baseado em uma lista de categorias de serviços oferecidos e os produtos desta categoria. Através destas informações o portal automaticamente gera uma interface web para comunicação entre o provedor e o consumidor de serviços.

O consumo de um serviço é realizado através das seguintes etapas:

1. Cidadão fornece suas credenciais ao portal (autenticação) ou cadastra-se como novo usuário;
2. O cidadão tem acesso a sua página personalizada, onde ele pode acessar os seus dados cadastrais, os detalhes dos serviços que ele já consumiu e localizar os serviços disponibilizados pelo portal, etc.
3. Quando realizar uma busca por um nome de uma categoria de um serviço, o portal apresenta todos os fornecedores registrados e uma lista de detalhes sobre o serviço desejado. Para cada provedor de serviço o portal apresenta outras informações relevantes na escolha do serviço como localização geográfica do provedor e uma espécie de pesquisa de satisfação de usuários que já negociaram com este fornecedor (reputação do provedor de serviço);
4. Quando escolher um produto ou um serviço atrelado a um fornecedor em particular, o portal automaticamente inicia o processo de negociação entre o consumidor e o

fornecedor. O portal comunica-se com o fornecedor através do *middleware*, seguindo as especificações detalhadas na seção 4.5;

5. Confirmado o consumo de um serviço pelo cidadão, o portal armazena em seu banco de dados todas as informações a respeito da negociação realizada;
6. Se desejar o consumidor pode visualizar o andamento do serviço consumido comunicando-se com o fornecedor através do portal relacional comunitário ou buscar novos serviços disponibilizados.

A utilização do portal relacional comunitário para cidades digitais pode ser exemplificada quando um usuário (cidadão) utiliza do ambiente das cidades digitais para comprar um ingresso de uma peça teatral em sua cidade.

O primeiro passo que o cidadão deve executar é acessar o portal relacional comunitário da sua cidade, que deve ser desenvolvido segundo as especificações da arquitetura das cidades digitais. Uma vez no portal relacional comunitário, o usuário deve buscar entre os serviços oferecidos, a peça teatral que deseja assistir. Selecionada a peça desejada, o portal direciona o usuário ao site de venda de ingressos comunicando-se através do *middleware*. Toda esta negociação é transparente ao portal relacional comunitário, pois se trata da implementação do serviço pelo provedor. Realizada a compra do convite, o provedor de serviços comunica-se com o portal informando que a transação foi bem sucedida. Em seguida, o portal registra as informações da transação e encaminha o usuário a página de busca de serviços, para que ele tenha acesso a outros serviços fornecidos pela cidade digital.

Após adquirir o ingresso do espetáculo, o usuário pode decidir se deseja contratar algum serviço de transporte ou qualquer outro serviço no portal da cidade digital.

Neste caso o usuário além de realizar a compra do ingresso, deseja alugar um carro e um traje de festa que será recebido em sua casa.

Com as entradas para a peça teatral compradas, o usuário opta por buscar um serviço de aluguel de carros. Ele seleciona a categoria “Aluguel de Carros” e busca empresas que disponibiliza este serviço.

Ao usuário é exibida uma lista com as empresas que disponibilizam esta categoria de serviço acrescidas de informações importantes para a escolha como por exemplo uma espécie de pesquisa de satisfação de usuários que já consumiram este serviço e localização

geográfica das empresas. Escolhida a companhia, é solicitada o aluguel de carros através de outro serviço disponibilizado pela cidade digital.

Realizada a transação o usuário é informado, através de uma propaganda, que será realizado um jantar beneficente no dia da peça teatral. Ele decide adquirir o convite para o jantar e para realização desta transação o usuário seguira os passos básicos da cidade digital, seleciona o setor seguido da empresa o produto e executa o pagamento.

Todos estes passos acima citados seriam normais para uma série de pessoas que tem por comum realizar transações via Internet. Porém neste caso o usuário não saberá que suas requisições feitas através do portal relacional comunitário desencadearão um grande número de requisições à diversos sistemas diferentes oferecidos pela cidade digital. Na realidade o usuário terá impressão de que não saiu da página do portal e que tudo foi feito através do portal relacional comunitário. Isto é possível devido ao alto grau de interoperabilidade gerado pela arquitetura das cidades digitais. A próxima seção descreverá a arquitetura proposta para desenvolvimento do portal relacional comunitário.

4.3 Arquitetura do Portal

A arquitetura desenvolvida é baseada no modelo MVC modificado (quatro camadas). Este modelo implica em poucos requisitos aos clientes (apenas um navegador web), criando assim uma independência de plataformas de utilização. [55]

O modelo MVC de quatro camadas proposto por Fowler [33] é uma evolução do modelo de três camadas. Ele retira a camada de apresentação do cliente e centraliza em um determinado ponto (geralmente um servidor Web). Desta forma o próprio cliente deixa de existir como um programa que precisa ser instalado em cada computador da rede. O acesso à aplicação é feito através de um navegador Web.

Todo o acesso do cliente ao banco de dados é feito de acordo com as regras de negócio contidas no servidor de aplicações. O cliente não tem acesso direto ao banco de dados, sem antes passar pelo servidor de aplicações. Sendo assim a segurança e controle de acesso ao banco de dados são feitos de maneira mais rígida e controlada.

Desta maneira a arquitetura do portal é composta por camada de apresentação, camada de aplicação, camada de negócios e camada de dados (Figura 12). Os clientes comunicam-se

apenas com a camada de apresentação, toda a lógica de negócios encontra-se no lado servidor.

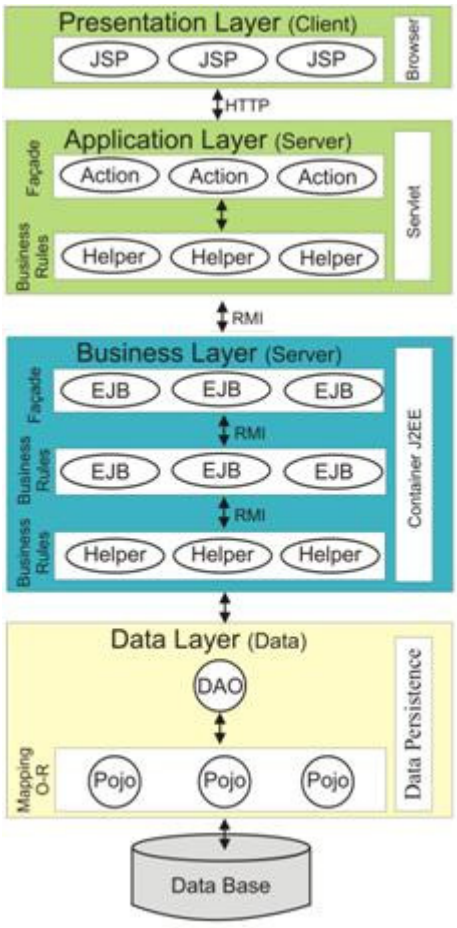

Figura 12 - Arquitetura de quatro camadas do Portal Relacional Comunitário [12]

As próximas seções descreverão resumidamente a função de cada camada no modelo MVC e as respectivas tecnologias utilizadas para a respectiva implementação.

4.3.1 Model (Negócios)

A camada de negócios representa os dados corporativos e regras de negócios que gerenciam o acesso e manipulação destes dados. Além disto, esta camada cria uma estrutura para implementação de distribuição, visando separação de conceitos e conseqüentemente fatores de qualidade como modularidade, extensibilidade e reusabilidade.

Na arquitetura proposta, a camada de negócios é implementada de forma a construir um sistema completo de gerenciamento através da utilização de componentes EJB[56][8]. Os componentes EJB's do tipo *Session Beans* ficam responsáveis por gerenciar o acesso concorrente aos serviços providos pelo sistema [57].

Os principais objetivos da tecnologia EJB são fornecer um rápido e simplificado desenvolvimento de aplicações Java baseado em componentes distribuídas, transacionais, seguras e portáveis.

Para definição dos EJB's é utilizada a API JEE, que em sua especificação define diversas funcionalidades que são oferecidas ao desenvolvedor. Dentre estas funcionalidades estão o gerenciamento transacional e controle de instâncias para requisições transparentes para o desenvolvedor, deixando o gerenciamento do ciclo de vida e controle transacional dos EJB's a cargo do Servidor de Aplicação. Estas vantagens trazem mais robustez para a camada de negócios, proporcionando assim escalabilidade [51].

Um servidor de aplicação é um software que disponibiliza um ambiente para a instalação e execução de aplicações. O objetivo de um servidor de aplicações é disponibilizar uma plataforma que abstraia do desenvolvedor de software algumas das complexidades de um sistema computacional. O servidor de aplicações responde a algumas questões comuns a todas as aplicações, como segurança, garantia de disponibilidade, balanceamento de carga e tratamento de exceções. Em um servidor de aplicação JEE[9] os módulos Web são implementados através de servlets e JSP e a lógica de negócio através de EJBs.

Alguns exemplos de servidores de aplicação são JBoss Application Server[58], GlassFish Application Server[59], IBM WebSphere Application Server[60], BEA WebLogic Application Server [61], Oracle Application Server [62].

[8] EJB ou Enterprise JavaBeans é um dos principais componentes da plataforma JEE (Java 2 Enterprise Edition, referenciar artigo JEE architecture). É um componente executado no container do servidor de aplicação JEE.

[9] JEE (Java EE, ou J2EE, ou Java 2 Enterprise Edition) é uma plataforma de programação para servidores na linguagem de programação Java.

4.3.2 View (Apresentação)

A camada View (Apresentação) mostra as informações recuperadas e processas pelas demais camadas e apresenta estas informações ao usuário através de uma interface. As tecnologias utilizadas para desenvolvimento desta camada são JSP (Java Server Pages) [63], JSF (Java Server Faces) [64] e Macromedia Flash [65].

4.3.3 Controller (Aplicação)

A camada *Controller*(aplicação) é responsável por despachar requisições e controlar seus fluxos. Na arquitetura proposta esta camada desempenha o papel de realizar a comunicação direta com o módulo responsável pela apresentação e entrada de dados de usuário [66].

Para a construção deste modelo, foi proposto a centralização da lógica de requisições, facilitando a implementação do controle de segurança e acesso ao sistema. Esta camada é implementada utilizando a linguagem Java.

4.3.4 Data Layer(Camada de Dados)

A arquitetura propõe que o acesso aos dados seja feito através de um conjunto de interfaces cuja finalidade é a de tornar mais simples a persistência e recuperação de modelos de objetos.

Este modelo possibilita a persistência de forma transparente dos modelos de objetos. O desenvolvimento ou a integração de bases heterogêneas se torna mais simples quando mudamos o foco da camada de persistência para a lógica do próprio aplicativo [67].

Para realizar a persistência de forma transparente, é utilizado o Mapeamento Objeto Relacional (ou ORM – Object Relational Mapping). ORM é uma técnica de desenvolvimento utilizada para visualizar um banco de dados relacional através dos paradigmas da programação orientada a objetos. As tabelas do banco de dados são representadas através de classes e os registros de cada tabela (tuplas) são representados como instâncias das classes correspondentes.

Como exemplo de *frameworks* que executam este mapeamento podemos citar o JPA [68] e o Hibernate[69].

Uma das vantagens da utilização do ORM é que o acesso a base de dados é feito através de classes e interfaces. Desta forma o desenvolvedor não precisará acessar o banco de dados via comandos SQL[10].

A persistência de dados nos permite agregar inúmeras facilidades e características interessantes à arquitetura proposta. Dentre elas, podemos destacar [67]:

- Demarcação de transações;
- Persistência por transitividade;
- Recuperação transparente (num dado grafo de objetos);
- Possibilidades de buscar objetos satisfazendo determinado critério;
- Facilidades de consulta;

A camada de dados tem como objetivo criar uma maneira inteligente de armazenar os dados através das facilidades provenientes da orientação a objetos.

A próxima seção descrever como o portal relacional comunitário se comunica com os sistemas dos provedores de serviço.

4.4 Comunicação com outros sistemas

Conforme visto na seção 3.2, depois de selecionado um serviço específico de um determinado provedor, o portal encaminha através do *middleware* a requisição ao provedor de serviços selecionado.

É de responsabilidade do provedor de serviços executar toda a transação eletrônica e informar o resultado desta negociação ao portal relacional comunitário.

A comunicação entre o portal e o provedor é realizada da seguinte forma:

1. Depois de selecionado o serviço e provedor específico, o portal encaminha (via *middleware*) os dados do usuário a *url* especificada no cadastro do provedor de serviços via XML definidos na seção 3.5 (Figura 13);
2. O provedor, através da sua loja-virtual, realiza todo o processo de compra e envio do produto/serviço aos dados cadastrais do usuário informados pelo portal relacional comunitário;
3. Realizada a transação, o provedor envia ao portal um XML contendo o resultado da transação e encaminha o cliente para página inicial do portal de serviços.

[10] SQL(Structured Query Language), é uma linguagem de pesquisa declarativa para banco de dados relacional

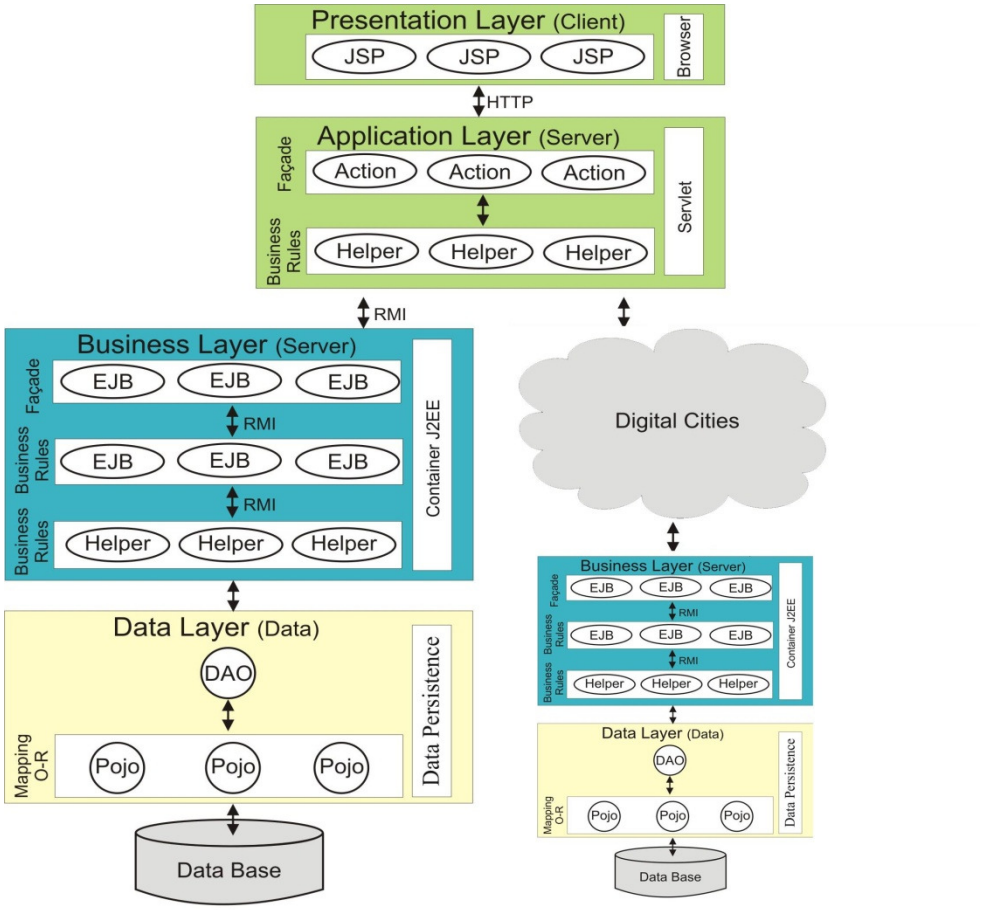

Figura 13 – Arquitetura de conexão do portal relacional comunitário

È importante salientar que toda esta comunicação é transparente ao usuário do portal. Esta transparência e comunicação de serviços são possibilitadas através da arquitetura apresentada na seção 3.3 e dos XML de comunicação que serão apresentados na próxima seção.

4.5 Padrões definidos

Com o objetivo de garantir interoperabilidade de todos os segmentos da sociedade, todos os setores devem estar engajados num esforço contínuo para assegurar que sistemas, processos e culturas de uma cidade real sejam gerenciados e direcionados para maximizar oportunidades de troca e reuso de informações e serviços. Com este objetivo, faz-se

necessária uma definição clara de políticas e especificações para interoperabilidade e gerenciamento de informação entre cidadãos, empresas, governo e provedores de serviço.

Estas políticas são especificadas através de estrutura de metadados (Xml). Segundo o W3C [44] estruturas XML foram projetadas para transportar e armazenar dados. Estruturas Xml podem ser utilizadas para as mais diversas finalidades, dentre elas podemos destacar compartilhamento e transporte de dados e mudanças de plataforma de hardware e software.

Usualmente, bancos de dados e sistemas de computação contêm dados em formatos incompatíveis. O armazenamento de dados nas estruturas XML é em formato de texto simples. Isto proporciona uma independência de software e hardware na persistência de dados. Desta forma com a utilização de estruturas XML, os dados podem ser facilmente trocados entre sistemas antes incompatíveis.

Um dos maiores desafios para os desenvolvedores de sistemas de informação é a troca de dados entre sistemas incompatíveis através da Internet [44]. Esta dificuldade apresenta-se especialmente em uma cidade digital onde muitos sistemas são desenvolvidos da forma mais heterogênea possível e estes necessitam realizar o intercâmbio de informações. Desta forma o uso de estruturas de metadados XML reduz esta complexidade, pois os dados podem ser lidos por diferentes aplicações e serviços antes incompatíveis. As próximas seções descreverão cada estrutura de Xml especificada para realizar o intercâmbio de dados em uma cidade digital.

1. Busca de categorias (Figura 14): este Xml é utilizado para requisição das categorias cadastradas no portal relacional comunitário. Baseada nesta lista um serviço pode classificar os seus produtos e registrá-los no portal;

```
¶
<?xml version="1.0" encoding="ISO-8859-1"?>¶
<digital_city_message>¶
»   <header>¶
»   »   <type>InfoEvent</type>¶
»   »   <from>Portal Comunitário</from>¶
»   </header>¶
»   <body>¶
»   »   <category>¶
»   »   »   <categoria>¶
»   »   »   »   <nome>Padaria</nome>¶
»   »   »   »   <description>Panificadoras da cidade digital</description>¶
»   »   »   </categoria>¶
»   »   »   <categoria>¶
»   »   »   »   <nome>Academia</nome>¶
»   »   »   »   <description>Academia da cidade digital</description>¶
»   »   »   </categoria>¶
»   »   </category>¶
»   </body>¶
</digital_city_message>¶
¤
```

Figura 14 - Xml de busca de categorias

2. Cadastro de produtos (Figura 15): esta estrutura é utilizada para cadastro e atualização dos produtos de um determinado fornecedor. Para realizar o cadastro de produtos o fornecedor deve estar previamente registrado no portal;

```
¶
<?xml version="1.0" encoding="ISO-8859-1"?>¶
<digital_city_message>¶
»   <header>¶
»   »   <type>EntryEvent</type>¶
»   »   <from>idPeer</from>¶
»   </header>¶
»   <body>¶
»   »   <product>¶
»   »   »   <category>Padaria</category>¶
»   »   »   <name>Produto A</name>¶
»   »   »   <quantity>10</quantity>¶
»   »   »   <price>1.00</price>¶
»   »   »   <stock>100</stock>¶
»   »   »   <terminate>false</terminate>¶
»   »   </product>¶
»   »   <product>¶
»   »   <category>Padaria</category>¶
»   »   »   <name>Produto B</name>¶
»   »   »   <quantity>2</quantity>¶
»   »   »   <price>13.00</price>¶
»   »   »   <stock>140</stock>¶
»   »   »   <terminate>false</terminate>¶
»   »   </product>¶
»   </body>¶
</digital_city_message>¶
¤
```

Figura 15 - Xml de cadastro de produtos

3. Transferência de usuário (Figura 16): este Xml é enviado a um fornecedor simultaneamente ao encaminhamento do cliente a página do serviço fornecido. Contém informações úteis sobre o serviço escolhido;

```
¶
<?xml version="1.0" encoding="ISO-8859-1"?>¶
<digital_city_message>¶
»   <header>¶
»   »   <type>SendUser</type>¶
»   »   <from>Portal Comunitário</from>¶
»   </header>¶
»   <body>¶
»   »   <user>¶
»   »   »   <idPeer>123456</idPeer>¶
»   »   »   <product>Pães</product>¶
»   »   </user>¶
»   </body>¶
</digital_city_message>¶
¤
```

Figura 16 – Xml de Transferência de usuário

4. Requisição de Informações de Usuário (Figura 17): possibilita a um *peer* requisitar informações ao portal relacional comunitário sobre um determinado usuário;

```
¶
<?xml version="1.0" encoding="ISO-8859-1"?>¶
<digital_city_message>¶
»   <header>¶
»   »   <type>InfoUser</type>¶
»   »   <from>idPeer</from>¶
»   </header>¶
»   <body>¶
»   »   <user>¶
»   »   »   <id_peer>idPeer</id_peer>¶
»   »   </user>¶
»   </body>¶
</digital_city_message>¶
¤
```

Figura 17 - Xml de Requisição de informações do usuário

5. Informações de um usuário (Figura 18): possibilita ao portal informar um provedor de serviços informações a respeito de um determinado usuário requisitar informações ao portal relacional comunitário sobre um determinado usuário;

```
¶
<?xml version="1.0" encoding="ISO-8859-1"?>¶
<digital_city_message>¶
»   <header>¶
»   »   <type>InfoUser</type>¶
»   »   <from>idPeer</from>¶
»   </header>¶
»   <body>¶
»   »   <user>¶
»   »   »   <id_peer>123456<id_peer>¶
»   »   »   <name>Eder</name>¶
»   »   »   <address>Rua Castro</address>¶
»   »   »   <cep>13100040</cep>¶
»   »   »   <city>Apucarana</city>¶
»   »   »   <uf>PR</uf>¶
»   »   »   <country>Brasil</country>¶
»   »   »   <tel>XXX</tel>¶
»   »   »   <email>xxx@xxx.com</email>¶
»   »   </user>¶
»   </body>¶
</digital_city_message>¶
¤
```

Figura 18 - Xml de informações do usuário

5 Modelagem e implementação do portal relacional comunitário

A modelagem e implementação do portal relacional comunitário definido no capítulo 4 é descrito detalhadamente neste capítulo. Esta seção apresenta a arquitetura do portal relacional comunitário dividido em uma série de visões: visão de casos de uso, visão lógica, visão de distribuição, visão dos dados e visão de implementação. Esta divisão da arquitetura por visões é inspirada na metodologia RUP [70] porém com algumas modificações como a inclusão da visão dos dados destinada a proporcionar uma melhor compreensão desta etapa do desenvolvimento. Em seguida são apresentados os detalhes e tecnologias utilizadas no desenvolvimento do modelo proposto. Por fim são apresentados alguns testes e resultados do portal implementado.

A modelagem do portal emprega a linguagem de modelagem de software UML [71] (*Unified Modelling Language)*. A UML é uma linguagem visual, que tem como objetivo fornecer uma maneira comum de capturar e expressar relações, comportamentos e idéias de um nível mais alto na forma de notações. A UML visa expressar a modelagem de um software da maneira mais simples e eficiente possível [72].

5.1 Visão de caso de uso

Esta seção contém uma descrição da visão de casos de uso da arquitetura de software. Na linguagem UML os diagramas de caso de uso são utilizados para capturar as funcionalidades e requisitos do sistema. Estes diagramas descrevem a interação entre um usuário e um sistema. Um diagrama de caso de uso apresenta o relacionamento entre um ator e um caso de uso. Os dois principais componentes de um diagrama de caso de uso são os casos de uso e os atores (Figura 19) [72].

Um ator é uma representação de um usuário ou outro sistema que irá interagir com o sistema que está sendo modelado. Um caso de uso é uma visão externa do sistema que representa uma ação que o usuário possa executar, a fim de completar uma tarefa [73].

De acordo com Fowler [73], diagramas de caso de uso devem ser usados na maior parte dos projetos de software, pois são úteis na obtenção de requisitos e planejamento do projeto.

Figura 19 - Componentes de um caso de uso

Os casos de uso do portal podem ser divididos em três atores principais. O cliente que utiliza o portal, provedores de serviço e o próprio portal relacional comunitário.

A Figura 20 apresenta o diagrama de casos de uso quando o ator é o usuário do portal relacional comunitário.

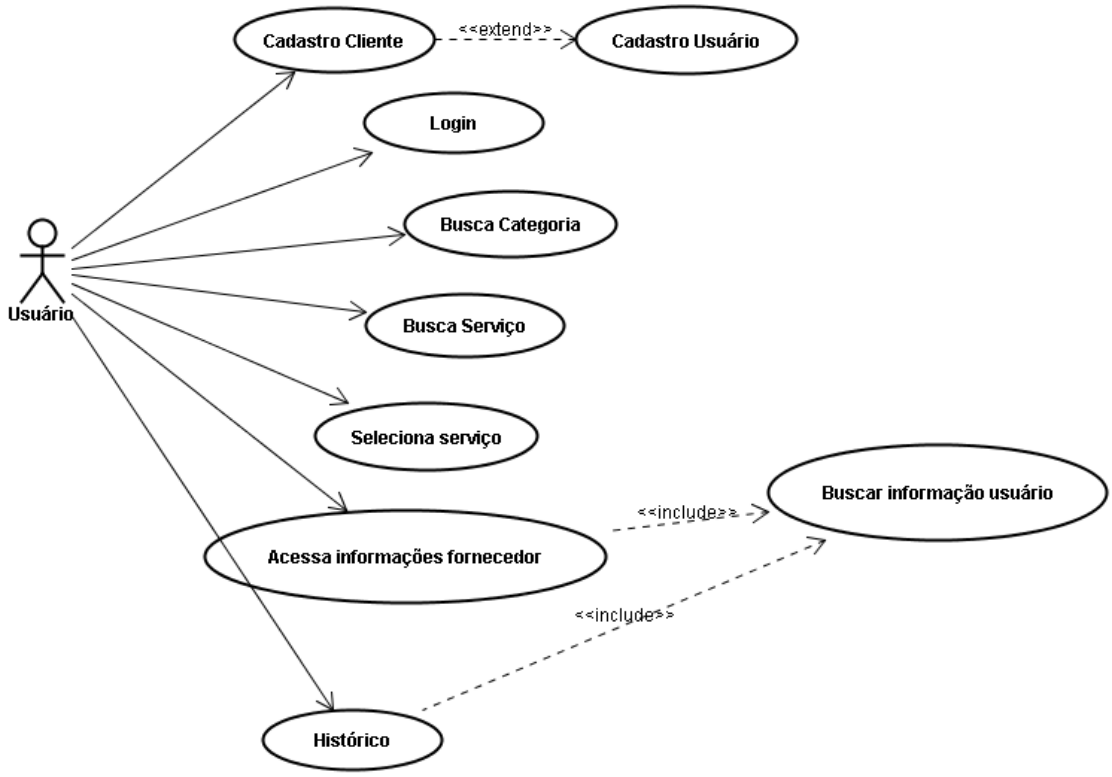

Figura 20 - Diagrama de Caso de Uso (Usuário)

A Figura 21 apresenta o diagrama de casos de uso referentes ao provedor de serviços.

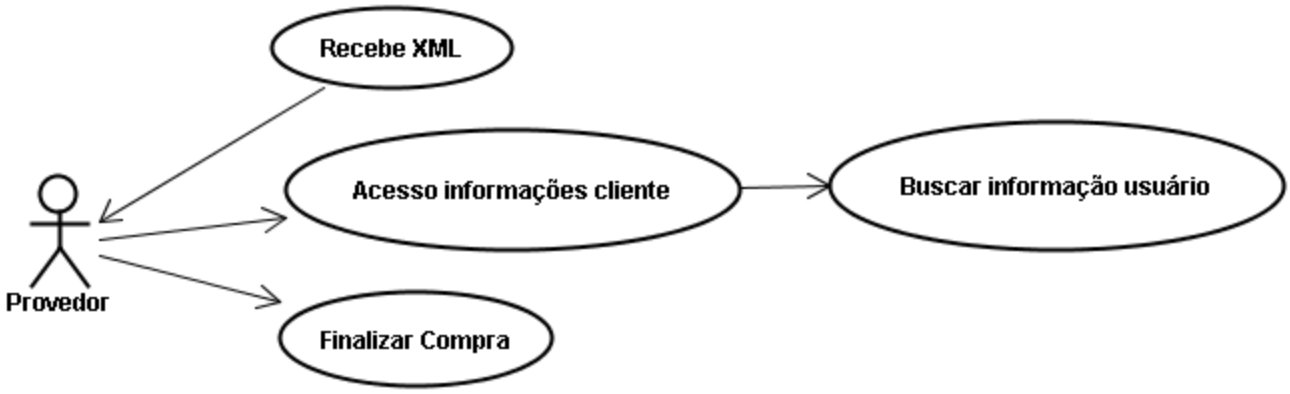

Figura 21 - Diagrama de Caso de Uso (Provedor)

A Figura 22 apresenta o diagrama de casos de uso referentes ao portal relacional comunitário onde um fornecedor envia informações ao portal referentes a um serviço.

Figura 22 - Diagrama de Caso de Uso (Portal)

5.2 Visão Lógica

Esta seção apresenta a visão lógica da arquitetura. Descreve as classes mais importantes, sua organização em pacotes e subsistemas de serviço e a organização destes subsistemas em camadas. Os diagramas foram incluídos para ilustrar os relacionamentos entre as classes, os subsistemas, os pacotes e as camadas arquiteturalmente significativas.

Como visto na seção 4.3 a visão lógica do Portal Relacional Comunitário é composta de quatro componentes principais (Figura 23): Apresentação, Aplicação, Negócios e Camada de Dados. O componente de apresentação depende da aplicação. Esta se comunica através de uma interface com o componente de negócios. O componente de negócios possui acesso a camada de Dados do sistema.

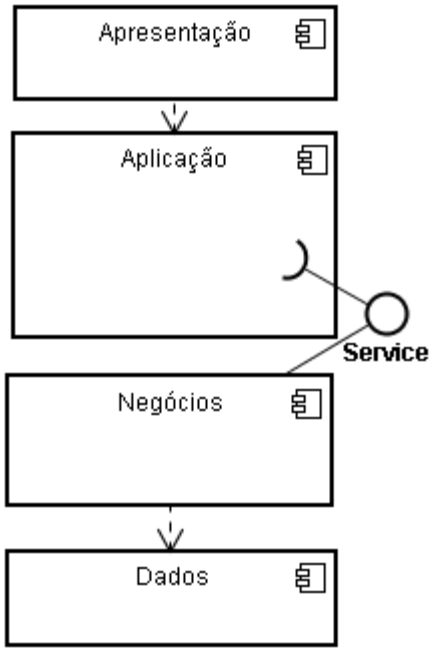

Figura 23 - Diagrama de Componentes

5.2.1 Diagramas de pacotes

O diagrama de pacotes, ou diagrama de módulos, descreve os pacotes ou pedaços do sistema divididos em agrupamentos lógicos criados com o intuito de caracterizar as dependências entre os pacotes. Estes diagramas ilustram a arquitetura do Portal mostrando o agrupamento de suas classes.

A Figura 24 apresenta o diagrama de pacotes padrão para a camada de apresentação do sistema. O pacote “unicamp.portal.jsp” agrupa classes que gerarão a página HTML exibida no navegador web do usuário. O pacote “unicamp.portal.managedBean” é o conjunto de classes que contém o estado do *“form”* e encaminham as requisições as classes do pacote “unicamp.portal.helper”.

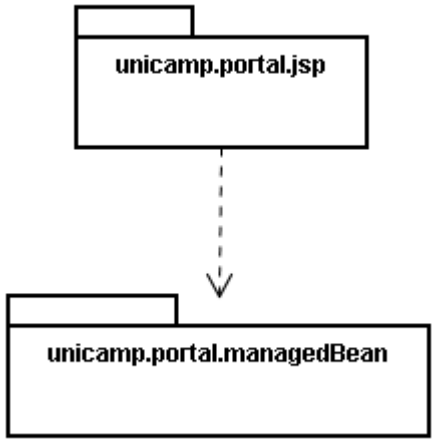

Figura 24 - Diagrama de Pacotes (Camada de Apresentação)

A Figura 25 apresenta o diagrama de pacotes padrão para a camada de aplicação do portal relacional comunitário. O pacote “unicamp.portal.helper” agrupa classes que são acessadas pelas classes do *managed bean* e contém toda a lógica da aplicação do portal relacional comunitário. Ela também é responsável por acessar a camada de negócios através de interfaces de comunicação.

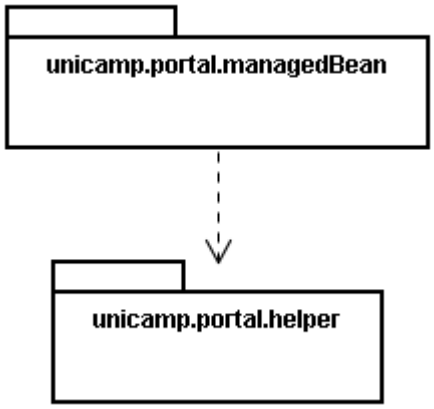

Figura 25- Diagrama de Pacotes (Camada de Aplicação)

O diagrama de pacotes para a camada de negócios é apresentado na Figura 26. O pacote “unicamp.portal.controller” contém as interfaces que serão disponibilizadas a camada de aplicação. O pacote “unicamp.portal.bo” contém a implementação de cada interface disponibilizada por esta camada. E o pacote “unicamp.portal.bdHelper” contém a respectiva implementação da lógica de negócio de cada módulo do sistema.

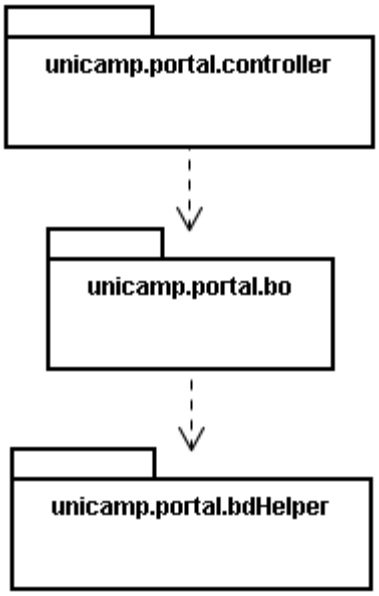

Figura 26 - Diagrama de Pacotes (Camada de Negócios)

A Figura 27 apresenta o diagrama para a camada de Dados do portal relacional comunitário. O pacote “unicamp.portal.dao” é o conjunto de toda a lógica de acesso ao banco de dados do sistema. O pacote “unicamp.portal.jpa” contém as classes de mapeamento objeto relacional do sistema. O pacote “unicamp.portal.pojo” contém classes utilitárias (objetos de negócio) criadas para enviar informações aos demais módulos do sistema.

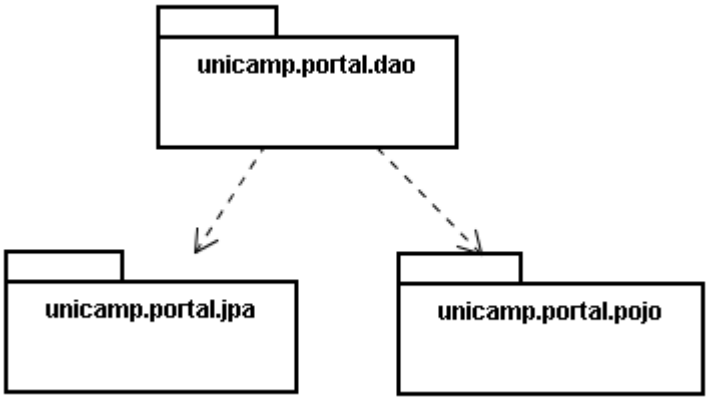

Figura 27 - Diagrama de Pacotes (Camada de Dados)

5.3 Visão de Distribuição

Esta seção descreve as configurações (hardware) e de rede física nas quais o portal será instalado e executado. Como foi utilizado JEE e a arquitetura de quatro camadas a instalação do portal relacional comunitário pode ser dividida em três servidores distintos (Windows ou Linux). Um contendo o servidor web, outro servidor atuando como servidor de aplicações e o terceiro servidor executando a função de banco de dados (Figura 28). Para acessar o portal relacional comunitário o cliente só precisa de um hardware (PC ou celular) que ofereça um navegador para acesso a Internet.

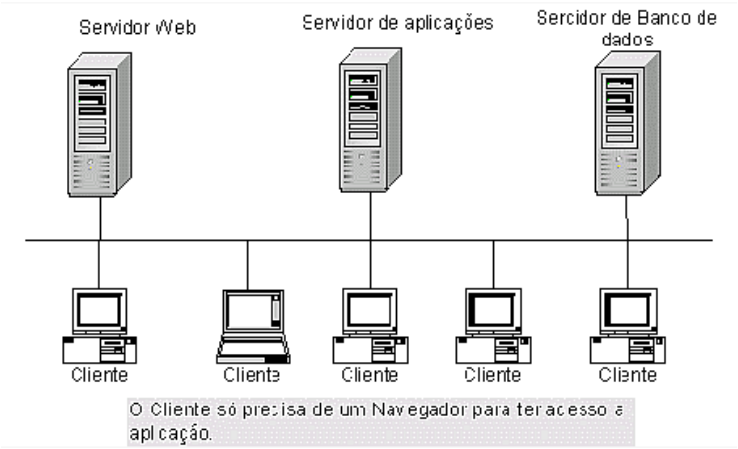

Figura 28 - Visão de distribuição

5.4 *Visão dos Dados*

Esta seção apresenta o modelo do banco de dados do portal relacional comunitário e a sua devida representação no mapeamento objeto relacional. A Figura 29 apresenta o modelo relacional do portal relacional comunitário.

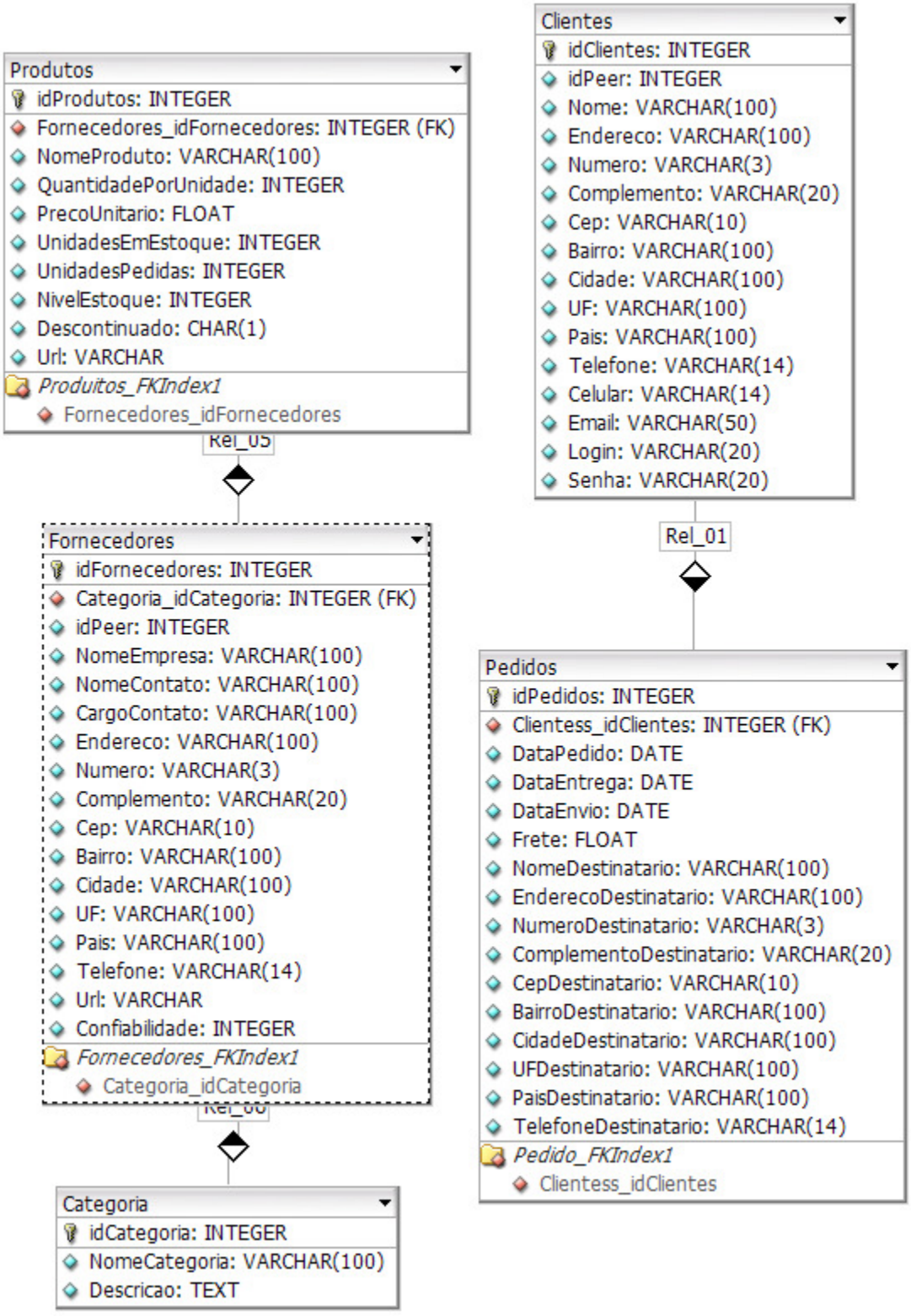

Figura 29 – Modelo Relacional

A tabela de clientes armazena informações referentes ao cadastro de pessoas no portal relacional comunitário. A tabela de categorias é utilizada para armazenar as categorias dos serviços oferecidos no portal relacional comunitário. Esta lista de categorias é uma das entidades chave na busca de serviços do portal.

A tabela de fornecedor contém informações referentes a fornecedores de serviço, as categorias que o fornecedor disponibiliza e contém uma relação com a tabela de produtos. Esta tabela registra o cadastro de produtos disponibilizados pelo provedor.

A tabela pedidos contém informações referentes aos pedidos realizados por um cliente no portal relacional comunitário e a tabela itens pedidos é criada para fornecer uma relação multi-direcional com a tabela de produtos.

O acesso ao banco de dados é feito através das técnicas de mapeamento objeto relacional conforme citado no capítulo 4.3. A implementação desta técnica é realizada através da especificação JPA da Sun [51]. A Figura 30 exemplifica como é realizado o mapeamento de uma tabela do banco de dados através de uma classe Java. Resumidamente o mapeamento é feito relacionando cada atributo da classe Java com a respectiva coluna da tabela no banco de dados.

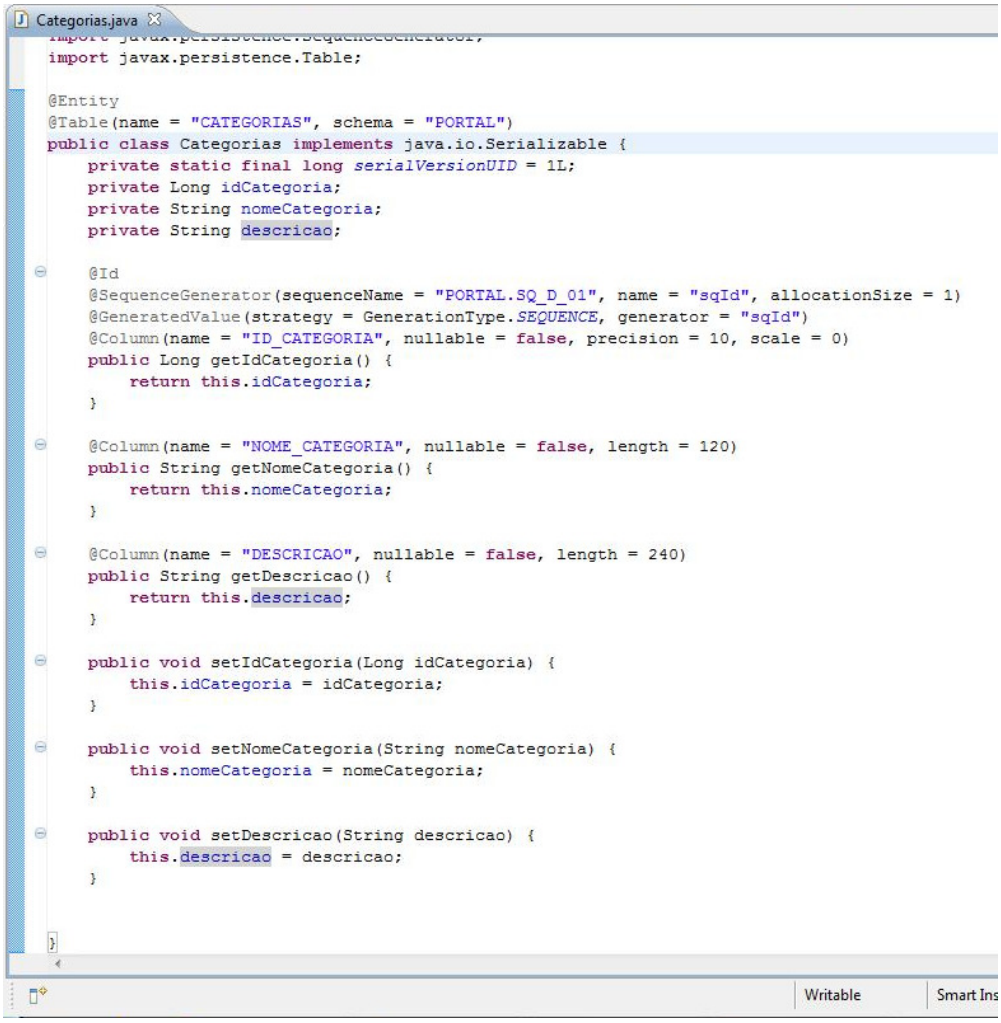

Figura 30 - Mapeamento JPA da tabela Categorias

5.5 Visão de Implementação

Esta seção descreverá a implementação de um caso de uso padrão no portal relacional comunitário. Este exemplo visa validar e exemplificar a utilização da arquitetura proposta. Em seguida será apresentado "*screen shots"* das principais funcionalidades do sistema. A Figura 31 apresenta o diagrama de seqüência para o caso de uso buscar categoria. A estrutura deste diagrama de seqüência pode ser replicada para a maior parte dos casos de uso do portal.

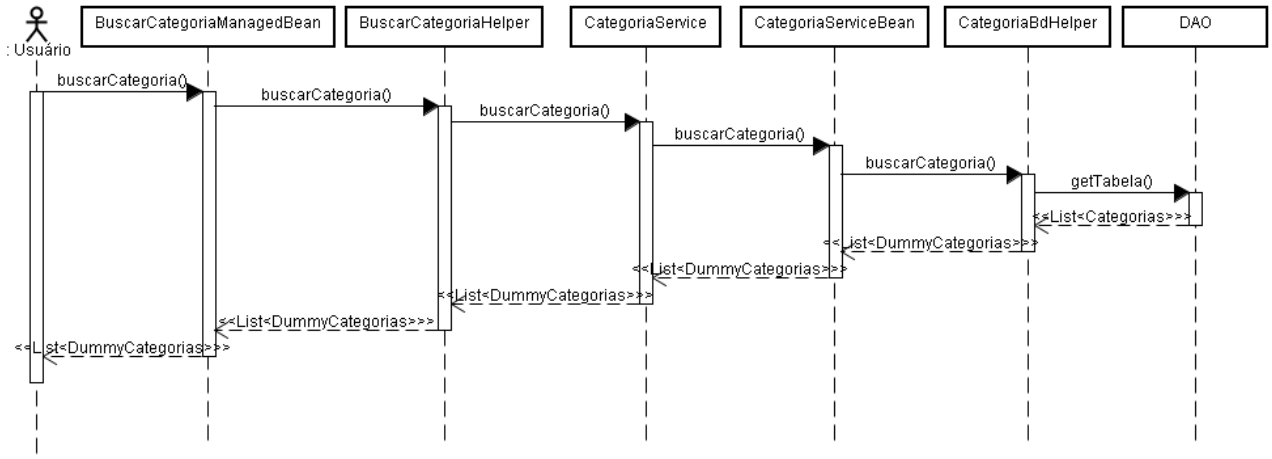

Figura 31 - Diagrama de Seqüência - Buscar Categoria

Neste caso de uso, o ator "Usuário" realiza uma busca por categoria. Ele seleciona a categoria "padaria" e clica no botão buscar (Figura 32). Em seguida, é instanciada a classe BuscarCategoriaManagedBean que encaminha a requisição a classe de lógica BuscarCategoriaHelper. Esta classe realiza a instanciação remota (via EJB) da interface CategoriaService. Esta interface é implementada pela classe CategoriaServiceBean. Esta classe encaminha a requisição para CategoriaBdHelper que realiza a consulta no DAO pela categoria desejada. Em seguida a classe CategoriaBdHelper copia o conteúdo do resultado para uma lista de POJOS[11] criando assim um objeto de transporte entre as diversas camadas da arquitetura e encaminha o resultado para a camada superior.

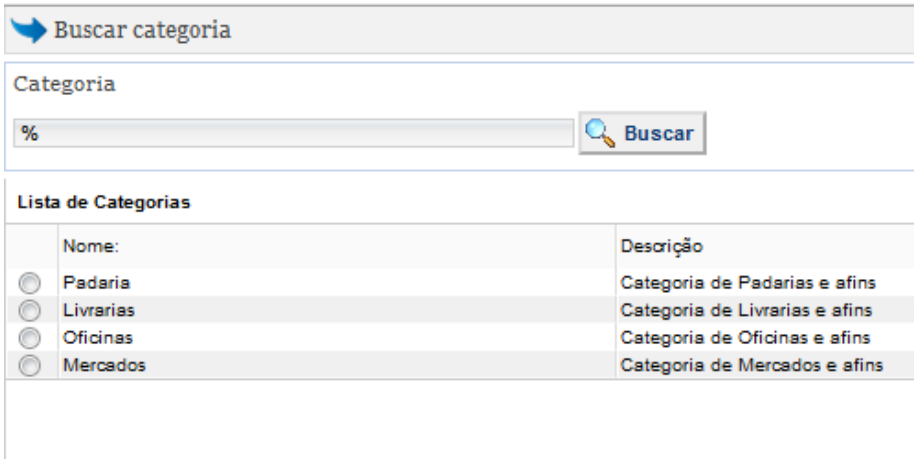

Figura 32 - Buscar Categoria

[11] Pojo(*Plain Old Java Objects)* são objetos Java que seguem uma estrutura simplificada usualmente contendo apenas métodos *setters* e *getters*.

A Figura 33 apresenta a tela de *login* do sistema, nela o usuário pode se autenticar no portal ou acessar a tela de Cadastro de Usuário.

Figura 33 – Login

Em seguida é apresentada ao usuário uma tela de abertura do portal relacional comunitário (Figura 34). Ela contém as aplicações que o usuário tem acesso no *menu* a esquerda da *interface.*

Figura 34 - Tela inicial

A busca e seleção por serviço é ilustrada na Figura 35 - Buscar Serviço e em seguida podemos visualizar a tela de informações do fornecedor. Ao confirmar a operação o usuário

é encaminhado ao provedor de serviços e o portal fica esperando a confirmação do consumo do produto selecionado.

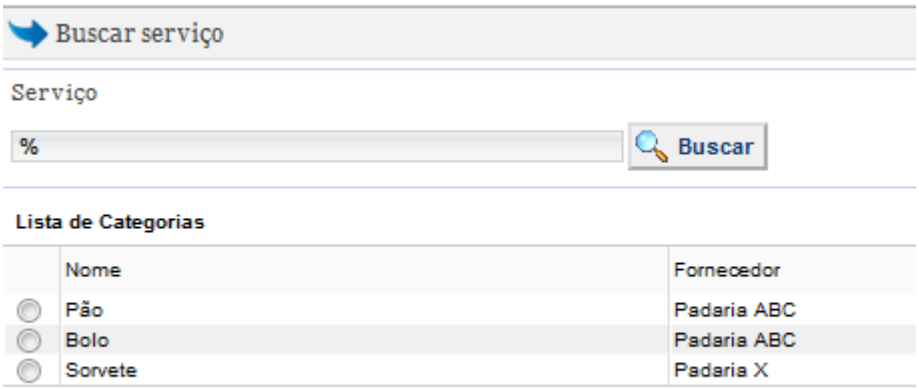

Figura 35 - Buscar Serviço

A próxima seção detalhará como foi realizado o processo de testes durante o desenvolvimento do protótipo do portal.

5.6 Testes

Para realização de testes no portal foi utilizada a abordagem do *Test Driven Development (TDD)* ou desenvolvimento orientado a testes. O TDD trata-se de uma prática de desenvolvimento de software onde o projeto do sistema se inicia a partir da unidade de teste de um objeto. Ela consiste de iterações curtas, escrevendo primeiramente casos de teste e depois implementando o objeto em questão.

De acordo com Chaplin[74], a principal regra de TDD é que “se você não pode escrever um teste para o objeto que você implementará, então você nunca deverá sequer pensar em codificar.”

Com a utilização do TDD, antes de escrever código de implementação, o desenvolvedor escreve casos de testes automatizados de unidade para a funcionalidade que será desenvolvida. Depois de escrever casos de teste (que geralmente não compilam), a etapa seguinte é implementar código para que o teste falhe. Um teste é considerado bem sucedido

quando este encontra uma falha na funcionalidade, logo quando um teste falha significa que para aquele caso de teste não foram encontrados falhas no sistema.

O ciclo então é repetido (escrita dos testes, implementação) com o desenvolvedor escrevendo alguns casos, implementando o código, e assim por diante. Desta forma, o trabalho é mantido dentro do controle intelectual do desenvolvedor, porque ele está continuamente realizando design e implementação de uma pequena funcionalidade [75]. A escrita de casos de teste para cada funcionalidade do sistema garante um nível "mínimo" de testes na aplicação facilitando o controle de mudanças no projeto e *refactoring*.

Como a preocupação com qualidade de código, controle de mudanças e *refactoring* é uma realidade na construção de um portal, o uso do TDD é uma iniciativa importante para assegurar o correto desenvolvimento desta aplicação.

Para implementação do TDD é utilizado o framework JUnit [76]. O JUnit é um *framework* que auxilia a criação e execução de testes unitários sendo assim útil para desenvolvimento do TDD. Ela possui suporte à criação de testes automatizados e estão integrada as principais IDES do mercado. A Figura 36 apresenta a utilização do TDD via JUnit na classe BuscarCategoriaHelper.

```java
package br.unicamp.testes.junit;

import static org.junit.Assert.assertEquals;

public class TestBuscaCategoriaHelper {

    public BuscarCategoriaHelper bHelper;

    @Before
    public void setUp() throws Exception {
        bHelper = new BuscarCategoriaHelper();
    }

    @Test
    public final void testBuscarCategoriaPorId() {
        PojoCategoria pojo = bHelper.buscarCategoriaPorId("1");
        assertNotNull(pojo);
        assertEquals(pojo.getNome(), "Padaria");
    }

    @Test
    public final void testBuscarCategoria() {
        List<PojoCategoria> lista = bHelper.buscarCategoria("p%");
        assertEquals(lista.isEmpty(), Boolean.FALSE);
        PojoCategoria pojo = lista.get(0);
        assertEquals(pojo.getNome().indexOf('p'), Boolean.TRUE);
    }

    @Test
    public final void testValidar() {
        // TODO
    }

}
```

Figura 36 - Classe de testes - BuscarCategoriaHelper

6 Conclusões e Trabalhos Futuros

Como visto neste trabalho a definição do termo Cidade Digital é divergente na literatura. Neste trabalho foi apresentado uma revisão bibliográfica de diversos assuntos referentes à cidades digitais. Além disto, foi realizada uma ampla discussão a respeito da definição precisa de uma cidade digital. Entende-se cidade digital como um espaço onde o cidadão pode divulgar suas idéias, negócios e realizar manifestações político-culturais das mais variadas formas possíveis. Também percebemos como principal fator motivador da construção de uma cidade digital o aprimoramento entre as relações entre o cidadão e o governo local.

O presente trabalho também propôs uma arquitetura para construção de cidades digitais. Nesta arquitetura a construção de cidades digitais é dividida em quatro grandes áreas (infra estrutura, middleware, arquitetura e aspecto organizacional), tendo como objetivo principal deste trabalho a criação de um modelo organizacional de intercâmbio de informações em uma cidade digital. Este modelo possibilita a criação de uma infra-estrutura de organização, disponibilização e consumo de serviços pelos cidadãos.

Para validar a proposta deste trabalho foi construído um portal comunitário para que o cidadão disponibilize e consuma serviços na arquitetura proposta. Este portal forneceu uma fácil disponibilização e rápida localização dos serviços oferecidos na cidade digital.

O presente trabalho atingiu os objetivos propostos no sentido que a arquitetura desenvolvida para cidades digitais e o modelo organizacional criado forneceu suporte efetivo a disponibilização e consumo de serviços ao cidadão através do portal comunitário.

Como trabalhos futuros sugerimos o estudo e aplicação de conceitos de web semântica no motor de buscas do portal. Desta forma pode-se proporcionar um melhor serviço aos cidadãos através do uso de uma linguagem natural na busca por serviços no portal comunitário.

Outra área para trabalhos futuros trata-se da criação de mecanismos de apoio a criação das lojas virtuais aos provedores de serviço. Estas se tratam dos sistemas que recebem as

requisições do portal comunitário e proporcionam o consumo efetivo do serviço pelos cidadãos.

Outra importante contribuição será a implantação de toda a arquitetura proposta e do portal comunitário em um *case* real. Como sugestão recomenda-se a implantação na Infovia de Pedreira [5] por tratar-se de um projeto já estabelecido e consolidado. Além disto, acredita-se na criação de uma pesquisa de satisfação dos usuários para identificar eventuais problemas e acolher sugestões dos cidadãos desta cidade.

7 Bibliografia

[1] Dicionário UOL – Michaelis, michaelis.uol.com.br/ Acessado em Setembro, 2009.

[2] Silva, Michéle T. Candido., 2002 A (Ciber) Geografia das Cidades Digitais. In Fluminense Federal University - Brazil.

[3] Ishida T., Isbister K. (Eds.), 2000, Digital Cities: Technologies, Experiences, and Future Perspectives, Springer-Verlag.

[4] Graham, Stephen., 2007, Rumo a Cidade em tempo real - Desenvolvimento Urbano numa Sociedade Globalizada e Telemediática. In http://www.wisetel.com.br/espaco_de_futuros/vcidade.htm - Brazil.

[5] Mendes, L.; Inocêncio A.; Panhan, A. Bringing Together Digital Cities and Open Access MANs, NAEC 2008

[6] André M. Panhan, Osman Lima, Leonardo S. Mendes , A WEB-BASED ARCHITECTURE FOR E-GOV APPLICATION DEVELOPMENT.

[7] Stott, B (1974). Review of load-flow calculation methods, IEEE Proceedings, vol. 62.

[8] Hagan, M. T. and Menhaj, M. B (1994). Training feed-forward networks with the Marquardt algorithm, IEEE Transactions on Neural Networks, vol.5, n.6.

[9] Paucar, V. L. and Rider, M. J (2002). Artificial Neural Networks for Solving the Power Flow Problem in Electric Power Systems, Electric Power Systems Research, vol.62, n.2.

[10] Panhan, A. M., Santos G. D, Mendes, L. S., Propostal of and Architecture for Digital Cities Creation

[11] Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides. Design Patterns: Elements of Reusable Object Oriented Software.Addison-Wesley, Reading – MA, 1994.

[12] Panhan, A. M., Ignatowicz E., Mendes, L. S., Community portals for architecture-based middleware P2P on IEEE - LATINCOM 2009.

[13] Lemos, A. http://www.guiadascidadesdigitais.com.br/site/pagina/o-que-cidade-digital. Acessado em Setembro, 2009.

[14] Graham II JW. Authenticating Public Access networking, SIGUCCS'02, 20–23 November 2002, Providence, Rhode Island, USA.

[15] Yamaoka J. 2006. Padrões de Interoperabilidade de Governo Eletrônico. In Reference document, the Federal Government of Brazil - Brazil.

[16] Dias, C. A. Dias Portal corporativo: conceitos e características http://www.scielo.br/scielo.php?pid=S0100-19652001000100007&script=sci_arttext&tlng=es . Acessado em Setembro, 2009.

[17] SHILAKES, Christopher C. & TYLMAN, Julie. Enterprise information portals. New York, NY: Merril Lynch, 16 Nov. 1998., outubro 1999. http://www.sagemaker.com/home.asp?id=500&file=Company/WhitePapers/lynch.htm. Acessado em Setembro, 2009

[18] METROPOLITAN AREA NETWORKS: HOW SOON? http://dome.mit.edu/ bitstream/handle/1721.3/37387/25086814. pdf?sequence=1. Acessado em Setembro, 2009

[19] Ming Xie, P2P System Based on Destributed Hash Table, Research Report for Seminar, University of Ottawa, Setembro, 2003.

[20] Redes Metropolitanas http://www.gta. ufrj.br/grad /05_1/redesopticas /Redes Metropolitanas. html. Acessado em Setembro, 2009

[21] MAN, http://standards.ieee.org/getieee802/802.11.html . Acessado em Setembro, 2009.

[22] Redes metropolitanas comunitárias http://repositorio .up .pt /aberto/bitstream/10216/12013/ 1/Resumo.pdf. Acessado em Setembro, 2009

[23] Miani, R. Aplicação de métricas à análise de segurança em Redes Metropolitanas de Acesso Aberto

[24] IBM Corporation, Glossary http://publib. boulder.ibm.com/infocenter/ pvcvoice/51x/ index.jsp?topic=/com. ibm.websphere.wvs.doc/wvs/glossary.html. Acessado em Setembro, 2009

[25] Rede Nacional de Ensino e Pesquisa, www.rnp.br/noticias/2006/not-060926.html. Acessado em Setembro, 2009

[26] Stanton http://www.rnp.br/wrnp2/2001/palestras_middleware/pal_middl_01.pdf

[27] Beverly Yang_ Hector Garcia-Molina, Improving Search in Peer-to-Peer Networks http://infolab.stanford.edu/~byang/pubs/p2psearch_long.pdf

[28] Peer-to-Peer Architecture Case Study: Gnutella Network http://cs-www.cs.yale.edu/homes/arvind/cs425/doc/gnutella-rc.pdf

[29] SETI@Home, http:// *setiathome.berkeley.edu/* . Acessado em Setembro, 2009

[30] Sousa, N. M. T., Peer-to-peer Computing. http://gnomo.fe.up.pt/~eol/MEMBERS/ nuno_sousa/old/ppc/artigo.html. Acessado em Setembro, 2009

[31] Nascimento. G - JXTA http://www.devmedia.com.br/articles/viewcomp.asp?comp=6053 . Acesso em setembro de 2009

[32] Carvalho, R. A. , Perspectivas na web semântica para a ciência da informação.

[33] Simon Sheu, Kien A. Hua and Wallapak Tavanapong - Chaining: A generalized batching technique for Video-on-Demand Systems

[34] Yang Guo, Kyoungwon Suh, Jim Kurose and DonTowsley - P2Cast: Peer-to-Peer patching scheme for VoD service

[35] Yang Guo, Kyoungwon Suh, Jim Kurose and DonTowsley A Peer-to-peer On-Demand streaming service and its performance evaluation

[36] Damiani E., Vimercati, D. , Paraboschi S. , A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks on Conference on Computer and Communications Security, 2002

[37] Lv, Q., Cao P., Cohen E, Search and replication in unstructured peer-to-peer networks on International Conference on Supercomputing, 2002

[38] Wang Y., Vassileva J. , Trust and Reputation Model in Peer-to-Peer Networks http://www.cs.usask.ca/grads/yaw181/publications/120_wang_y.pdf Acessado em Setembro, 2009

[39] Li gong , Project JXTA: A Technology Overview http://admis.fudan.edu.cn/member/yztang/Papers/P2P/industry/JXTA%20TechOverview.pdf . Acessado em Setembro, 2009

[40] Sun Microsystems. The Source for Java Developers. http://java.sun.com. Acesso em julho de 2009.

[41] CiteSeerX — *Project JXTA* 2.0 Super-Peer *Virtual Network*, Project JXTA Virtual Network Traversat, http: *citeseer.ist.psu.edu/675705.html .* Acessado em Setembro, 2009

[42] Bisignano, http://zeus.ws.dei.polimi.it/is-manet/Documenti/pap-diit-2.pdf . Acessado em Setembro, 2009

[43] Answers Dictionary - http://www.answers.com/topic/peer-to-peer , Acesso em setembro de 2009.

[44] World Wide Web Consortium http://www.w3.org/2001/sw/SW-FAQ# Acesso em setembro de 2009.

[45] Scientific American – The Semantic Web . http:// www. si.umich.edu/~rfrost/ courses/si110/readings/in_out_and_beyond/semantic_web.pdf Acesso em setembro de 2009.

[46] Oliveira, R. M. V. B. WEB SEMÂNTICA: NOVO DESAFIO PARA OS PROFISSIONAIS DA INFORMAÇÃO - http://www.sibi.ufrj.br/snbu/snbu2002/oralpdf/124.a.pdf . Acesso em setembro de 2009.

[47] World Wide Web Consortium FAQ - http://www.w3.org/2001/sw/sweo/public/UseCases/ Acesso em setembro de 2009.

[48] World Wide Web Consortium FAQ http://www.w3.org/2001/sw/sweo/public/UseCases/ Zaragoza/ Acesso em setembro de 2009.

[49] World Wide Web Consortium FAQ http://www.w3.org/2001/sw/sweo/public/ UseCases/CTIC/ Acesso em setembro de 2009.

[50] Théodoloz N., 2004. DHT-Bases Routing and Discovery in JXTA. In School of computer and Comunication Sciences – Computer Sciences departement – École Polytechnique Fédérale de Lausanne - Switzerland.

[51] Java EE at a Glance. java.sun.com/javaee/ Acessado em Setembro, 2009.

[52] ICP-Brasil, http://www.iti.gov.br/ . Acessado em Setembro, 2009.

[53] Robert Flener at all, Java P2P unleashed, Editora Sams, first edition, Indianapolis – USA, agosto, 2002.

[54] Nicolas Théodoloz, DHT-Bases Routing and Discovery in JXTA, Diploma Project, School of computer and Comunication Sciences – Computer Sciences departement – École Polytechnique Fédérale de Lausanne – Suiça, Fevereiro, 2004.

[55] Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides. Design Patterns: Elements of Reusable Object Oriented Software. Addison-Wesley, Reading – MA, 1994.

[56] EJB, http://java.sun.com/products/ejb/. Acessado em Setembro, 2009.

[57] Ivar Jacobson. Object-Oriented Software Engineering. Addison - Wesley Longman, Inc. 1992.

[58] JBoss Application Server, http:// www.jboss.org/. Acessado em Setembro, 2009.

[59] GlassFish Application Server, https://glassfish.dev.java.net/. Acessado em Setembro, 2009.

[60] IBM WebSphere Application Server, www.ibm.com/websphere. Acessado em Setembro, 2009.

[61] BEA WebLogic Application Server, www.oracle.com/bea/index.html, Acessado em Setembro, 2009.

[62] Oracle Application Server, www.oracle.com/appserver/index.html, Acessado em Setembro, 2009.

[63] Java Server Pages Technology - java.sun.com/products/jsp/. Acessado em Setembro, 2009.

[64] Java Server Faces - java.sun.com/javaee/javaserverfaces/. Acessado em Setembro, 2009.

[65] Adobe Flash - www.adobe.com/br/products/flashplayer/ Acessado em Setembro, 2009.

[66] Dan Pilone and Neil Pitman. UML 2.0 in a nutshell. O'Reilly Media, Inc., 1 edition, 6 2005.

[67] Deepak Alur, John Crupi, Dan Malks. Core J2EE Patterns. Sun Microsystems, Inc. – 2001.

[68] JPA - Java Persistence API, http:// java.sun.com/javaee/overview/faq/persistence.jsp. Acessado em Setembro, 2009.

[69] Hibernate - https://www.hibernate.org/. Acessado em Setembro, 2009.

[70] Chaplin, D., "Test First Programming," TechZone, 2001.

[71] OMG. Unified Modeling Language. http://www.uml.org/. Acesso em setembro de 2009.

[72] Dan Pilone and Neil Pitman. UML 2.0 in a nutshell. O'Reilly Media, Inc., 1 edition, 6 2005.

[73] Martin Fowler, Kendall Scott: UML Distilled Addison-Wesley 2000

[74] Chaplin, D., "Test First Programming," TechZone, 2001.

[75] An Initial Investigation of Test Driven Development in Industry George B., Willians L. on ACM 2003.

[76] JUnit http://www.junit.org/