

**Rogeria Ramos de Oliveira Monteiro**

**PORTAL PESSOAL UNIVERSAL**

Florianópolis – SC  
2002

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**Rogéria Ramos de Oliveira Monteiro**

**PORTAL PESSOAL UNIVERSAL**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Prof<sup>ª</sup>. Elizabeth Sueli Specialski, Dr<sup>ª</sup>.

Florianópolis, Agosto de 2002.

# PORTAL PESSOAL UNIVERSAL

Rogeria Ramos de Oliveira Monteiro

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

---

Prof. Fernando Ostuni Gauthier, Dr.  
Coordenador do Curso

Banca Examinadora

---

Prof<sup>a</sup>. Elizabeth Sueli Specialski, Dr<sup>a</sup>.  
Orientadora

---

Prof. Fernando Ostuni Gauthier, Dr.

---

Prof. João Bosco da Mota Alves, Dr.

“Determinação, coragem e autoconfiança são fatores decisivos para o sucesso. Não importa quais sejam os obstáculos e as dificuldades. Se estamos possuídos de uma inabalável determinação, conseguiremos superá-los.”

Dalai Lama

Agradeço à Prof<sup>a</sup> Elizabeth Specialski, pela confiança e por ter me dado a honra de compartilhar de seu conhecimento e de sua amizade, aos professores componentes da banca examinadora, pela avaliação e colaborações, e também, à Universidade Federal de Santa Catarina.

Sou muito grata a Deus, por ter me oferecido esta oportunidade e pela grande força que me concede diariamente.

Obrigada às minhas famílias pelo incentivo, dedicação e paciência, assim como a todos que colaboraram de alguma forma para a realização desta dissertação.

# Sumário

Lista de Figuras .....	viii
Lista de Quadros .....	x
Lista de Tabelas .....	xi
Resumo .....	xii
Abstract.....	1
1 Introdução.....	1
1.1 Considerações Iniciais .....	1
1.2 Objetivos.....	4
1.2.1 Objetivo Geral .....	4
1.2.2 Objetivos Específicos .....	4
1.3 Justificativa.....	5
1.4 Limitações .....	5
1.5 Estrutura do Trabalho .....	6
2 Aspectos de desenvolvimento de um <i>site</i> .....	7
2.1 Planejamento .....	7
2.2 Construção.....	10
2.3 Segurança.....	15
2.3.1 Criptografia.....	16
2.3.2 Função <i>Hash</i> .....	17
2.3.3 Assinatura Digital .....	18
2.3.4 Certificado Digital .....	19
2.3.5 Infra-estrutura de chaves públicas - ICP.....	21

2.3.6 Protocolização de documentos eletrônicos.....	22
2.4 Organização e Armazenamento das Informações .....	23
2.4.1 O Modelo de dados Orientado a Objetos.....	23
2.4.2 O Paradigma Relacional Orientado a Objetos .....	30
2.5 Modelagem Lógica dos Dados .....	41
2.5.1 Definição dos Requisitos .....	42
2.5.2 Casos de Uso – Uses case.....	43
2.5.3 Modelo Conceitual .....	45
2.5.4 A Ferramenta Glossário.....	47
2.5.5 Diagrama de Seqüência do Sistema.....	47
2.5.6 Diagrama de Interação.....	48
2.5.7 Diagrama de Classes.....	50
2.5.8 Pacotes .....	53
2.5.9 Diagrama de Estados .....	53
2.5.10 Diagrama de Atividades .....	54
2.5.11 Diagrama Físico.....	57
2.6 Agentes de <i>Software</i> .....	58
2.6.1 Definição .....	58
2.6.2 Propriedades .....	59
2.6.3 A Sociedade de Agentes .....	61
2.6.4 Classificação .....	62
3 Modelo Proposto .....	63
3.1 Sessão Pessoal .....	65
3.2 Sessão Profissional .....	65
3.3 Sessão Saúde .....	68
3.4 Sessões de Demanda e de Oferta.....	69

3.4.1 Agente de Compra – Sessão de Demanda .....	71
3.4.2 Agente de Venda – Sessão de Oferta.....	74
4 O Portal Pessoal Universal .....	79
4.1 Sessão Pessoal .....	86
4.2 Sessão Profissional .....	88
4.3 Sessão Saúde .....	89
4.4 Sessão de Demanda .....	90
4.5 Sessão de Oferta .....	91
5 Considerações Finais .....	92
6 Referências Bibliográficas.....	95
6.1 Bibliografias .....	95
6.2 <i>Sites</i> Consultados.....	98
Anexo 1 - Lista de Categoria de Conceitos .....	100
Anexo 2 – Medida Provisória 2200-2 .....	101
Anexo 3 - Glossário.....	106



## LISTA DE FIGURAS

Figura 1.1: Modelo de Site Pessoal Universal proposto por David Siegel.....	3
Figura 2.1: Exemplo de hipertexto .....	10
Figura 2.2: Modelo linear de navegação .....	11
Figura 2.3: Modelo hierárquico de navegação .....	11
Figura 2.4: Modelo raio e centro de navegação.....	12
Figura 2.5: Modelo web designer completo de navegação .....	13
Figura 2.6: Assinatura Digital .....	18
Figura 2.7: Processo de Emissão de Certificado Digital .....	21
Figura 2.8: Modelo Conceitual do “Sistema Controle de Veículos” .....	46
Figura 2.9: Diagrama de Seqüência do Sistema para o Caso de Uso Cadastro de Veículos .....	47
Figura 2.10: Nomenclatura e simbologia parcial de classes e instâncias ou objetos ....	48
Figura 2.11: Diagrama de Seqüência para o Caso de Uso Entrada de Veículos no Estoque .....	49
Figura 2.12: Diagrama de Colaboração para o Caso de Uso Entrada de Veículos no Estoque .....	50
Figura 2.13: Nomenclatura e simbologia UML de uma Classe .....	50
Figura 2.14: Diagrama de Classes do “Sistema Controle de Veículos” .....	52
Figura 2.15: Nomenclatura e simbologia de Pacotes .....	53
Figura 2.16: Diagrama de Estados para um Veículo .....	54
Figura 2.17: Diagrama de atividades para a Entrada de veículos no estoque .....	56
Figura 2.18: Diagrama de Componentes combinado com o Diagrama de Utilização ...	57
Figura 3.1: Representação das Sessões e dos Níveis de Acesso, com sugestão de informações a serem armazenadas no <i>Site</i> Pessoal Universal. ....	64

Figura 3.2: Processo de compra e venda com Agentes Facilitadores.....	71
Figura 4.1: O modelo do Portal Pessoal Universal.....	79
Figura 4.2: Diagrama de Atividades do Portal Pessoal Universal.....	81
Figura 4.3: Diagrama de Atividades do Construtor de <i>Sites</i> Pessoais Universais - CSPU .....	81
Figura 4.4: Diagrama de Atividades da Inclusão de Sessões .....	82
Figura 4.5: Diagrama de Atividades da Navegação em um Site Pessoal.....	82
Figura 4.6: Página inicial do Portal Pessoal Universal.....	85
Figura 4.7: Elaboração da página inicial do <i>Site</i> Pessoal no Construtor de <i>Sites</i> Pessoais Universais .....	86
Figura 4.8: Formulário da Sessão Pessoal .....	87
Figura 4.9: Solicitação de nova informação na Sessão Pessoal .....	87
Figura 4.10: Formulário da Sessão Profissional.....	88
Figura 4.11: Formulário da Sessão Saúde. ....	89
Figura 4.12: Formulário da Sessão Quero, exemplificando o preenchimento do formulário para procurar por uma babá.....	90
Figura 4.13: Formulário da Sessão Tenho, demonstrando o preenchimento do descritor de uma bicicleta. ....	91

## LISTA DE QUADROS

Quadro 2.1: Declaração da relação Veículos utilizando XSQL através da definição de tipos .....	33
Quadro 2.2: Declaração da relação Veículos utilizando XSQL sem a criação de tipos.	34
Quadro 2.3: Declaração do tipo Pessoa.....	34
Quadro 2.4: Herança de tipos .....	34
Quadro 2.5: Herança múltipla de tipos.....	35
Quadro 2.6: Renomeação de atributos com mesmo nome na aplicação de herança múltipla de tipos. ....	35
Quadro 2.7: Herança de tabelas.....	36
Quadro 2.8: Herança múltipla de tabelas.....	36
Quadro 2.9: Representação de tipos complexos.....	37
Quadro 2.10: Tipo Acessório e Tabela VeículosObj.....	38
Quadro 2.11: Consulta sobre atributo relação-valorado.....	38
Quadro 2.12: Consulta sobre atributo relação-valorado na SQL extendida.....	38
Quadro 2.13: Consulta sobre atributo relação-valorado na SQL extendida.....	38
Quadro 2.14: Expressões path .....	39
Quadro 2.15: Atribuição de valores aos objetos através da instrução <i>Insert Into</i> .....	40
Quadro 2.16: Requisitos básicos de um sistema .....	43
Quadro 2.17.a: Caso de uso Cadastro de Veículos.....	44
Quadro 2.17.b: Caso de uso Entrada de veículos no estoque.....	45
Quadro 3.1: Demonstração de criação da tabela TENHO e de uma consulta na linguagem XSQL feita por um Agente de Compra em busca de uma bicicleta.....	74

## **LISTA DE TABELAS**

Tabela 2.1: Representação da relação Veículos .....	32
Tabela 2.3: Representação da relação Veículos apresentando os dados incluídos com a operação Insert Into .....	40

## **RESUMO**

Este trabalho apresenta o projeto de um Portal Pessoal Universal, cujo objetivo é centralizar o acesso a *Sites* Pessoais que organizam informações relativas a uma pessoa, num formato padrão, a ser usado universalmente como referência. As informações sugeridas são divididas em 5 sessões – Pessoal, Profissional, Saúde, Quero e Tenho – que possuem um controle de acesso distribuído em 6 níveis devidamente assegurados.

Um embasamento teórico é apresentado sobre a construção de *sites* na Internet, onde o planejamento, a organização e a disponibilização de seu conteúdo são evidenciados. Em acréscimo, são feitas considerações sobre a necessidade de proteção das informações e sobre a complexidade de se implantar diversos níveis de acesso, sugerindo-se a protocolização digital de dados e de documentos, além do uso de certificados digitais.

## **ABSTRACT**

This work presents the project of Universal Personal Portal, with goal to meet the access to the Personal Sites which organize information concerning a person using a pattern form that used universally how reference. The suggested information are divided in five sessions – Personal, Professional, Health, I Want and I Have – under an access control system, distributed among six levels with security guarantee.

A theoretical fundamentals about the sites's construction in the internet is presented, where the planning, the organization and the distribution of the contents are evident. In addition, the information protection necessity and the complexity to implant several security access levels are explained and discussed. As a solution, the data and document register and the digital certification is suggested.

# **1 INTRODUÇÃO**

## **1.1 Considerações Iniciais**

Na atualidade verifica-se claramente a ocorrência de uma nova revolução mundial. É a revolução tecnológica guiada pela informação, resultando em mudanças profundas na maneira de administrar empresas e países. Novas estratégias precisam ser elaboradas, pois a informação passa a ser um objeto de fundamental valor, circulando a velocidade da luz através de fibras óticas, satélites e linhas telefônicas. “Isto gera implicações técnicas, sociais, culturais, políticas e econômicas irreversíveis, que mudam as formas de discutir e organizar a sociedade” (Livro Verde da Sociedade da Informação..., 2000), fazendo surgir uma nova geração conhecida por Sociedade da Informação.

Constata-se que o gerenciamento e acesso à informação, principalmente à científica e tecnológica, passaram a ser considerados requisitos fundamentais para o progresso, pois a tecnologia da informação é a principal ferramenta para obter e dar continuidade a ganhos significativos de produtividade que estarão moldando os líderes deste século (OLIVEIRA, 2000). A administração, disponibilidade, precisão e relevância da informação têm constituído a base de processos decisórios, logo é preciso criar e manter uma infra-estrutura de informação capaz de viabilizar o processo de produção de conhecimento e sua utilização como subsídio ao desenvolvimento.

Diante disso, é preciso o auxílio do governo para criar condições onde ocorra crescente confiança pública nesse novo contexto social, iniciando com investimento na área educacional, alterando a metodologia hoje aplicada, para que os cidadãos participem da Sociedade da Informação desde a infância, crescendo aptos a analisar e criticar as informações que recebem (OLIVEIRA, 2000). Sendo assim, com profissionais de qualidade, vários setores do país seriam induzidos a saltos qualitativos e quantitativos colocando a nação no cenário internacional e tornando seu povo mais rico, culto e justo (Ciência e Tecnologia para a Construção..., 1999). Mais adiante nesse

processo, os referenciais de proteção à privacidade individual, assim como de segurança no armazenamento e no fluxo de informações comerciais, financeiras, científicas e tecnológicas, também precisam da participação e investimento conjuntos de organizações públicas e privadas.

O crescimento do número de computadores interligados pela Internet tem sido exponencial. Num curto período de oito anos, a Internet se disseminou por praticamente todo o mundo, propiciando conectividade a países que não utilizavam redes e substituindo outras mais antigas. Mesmo sendo ainda um serviço restrito a poucos em vários países, a velocidade de disseminação da Internet, comparada com outros serviços, mostra que ela se tornou um “padrão de fato e que todo indivíduo vive diante de um fenômeno singular, a ser considerado como fator estratégico fundamental para o desenvolvimento das nações” (Livro Verde da Sociedade da Informação..., 2000). Contudo, é preciso fazer com que sejam realizados meios de levar a informação de forma igualitária a todo cidadão e que estes possam ser beneficiados pela inclusão do nosso país na Sociedade da Informação.

Verifica-se que a tecnologia da informação diversificou-se para satisfazer as demandas da sociedade. A grande maioria dos profissionais de informática está migrando seus trabalhos para o ambiente Internet, porque acreditam em seu sucesso. A indústria da tecnologia da informação defende a padronização e uniformização nos modelos adotados para o desenvolvimento e distribuição de conteúdo, a fim de haver mais integração entre as soluções de tecnologia. Afinal, é de grande importância que uma aplicação possa ser utilizada sem restrição de local e horário (OLIVEIRA, 2000).

No Brasil, pode-se considerar que, uma vez dominados os processos e métodos básicos da Internet – como o estabelecimento de uma base para a criação de uma infraestrutura de rede no país, a instalação do *backbone* nacional e o incentivo ao desenvolvimento de serviços, aplicações e conteúdos para a rede brasileira, através da RNP/MCT (Rede Nacional de Pesquisa / Ministério de Ciência e Tecnologia) – “deve-se agora partir para uma nova fase ‘orientada para aplicações’, onde ‘o que’ fazer, usando os serviços e infra-estrutura instalados, torna-se muito mais importante do que a rede em si” (Ciência e Tecnologia para a Construção..., 1999).



Logo, “é importante definir o contexto em que convivem duas redes: uma que está sendo usada ‘na prática’ e outra, que está sendo desenvolvida para ser usada ‘no futuro’” (ibidem). A nova fase da Internet necessita de velocidade e qualidade de serviço não encontradas na atual infra-estrutura para que seja viável a implantação de aplicações mais avançadas. Pensa-se em serviços realmente úteis, que façam diferença nos compromissos diários de cada cidadão.

Siegel (SIEGEL, 2000) faz uma previsão de como o mundo poderia ser em 2010, onde um *Web Site* Pessoal é uma questão de necessidade e um direito de todos, pois através dele, uma pessoa pode administrar sua vida com a máxima segurança existente através da Internet. Em seu modelo apresentado na Fig. 1.1, diversas atividades passam a ser viáveis através desse espaço, entre elas pode-se citar a obtenção de todo tipo de documento, a compra e venda de produtos/serviços com o auxílio de descritores universais, o armazenamento de chaves eletrônicas de vários lugares que se tem acesso, a procura e admissão de funcionários, a organização de um inventário de todos os bens da família, a disponibilização de um prontuário eletrônico para auxiliar nas questões de saúde, o ingresso em colégios e universidades, além da utilização de diversos serviços com a manipulação do dinheiro eletrônico. As informações são organizadas em 7 (sete) sessões – Pessoal, Profissional, Financeira, Médico, Governo, Tenho e Quero – e disponibilizadas em seis níveis de privacidade e de segurança – Público, Emergencial, Conhecidos e Colegas, Amigos e Família, Cônjuge e O Titular.

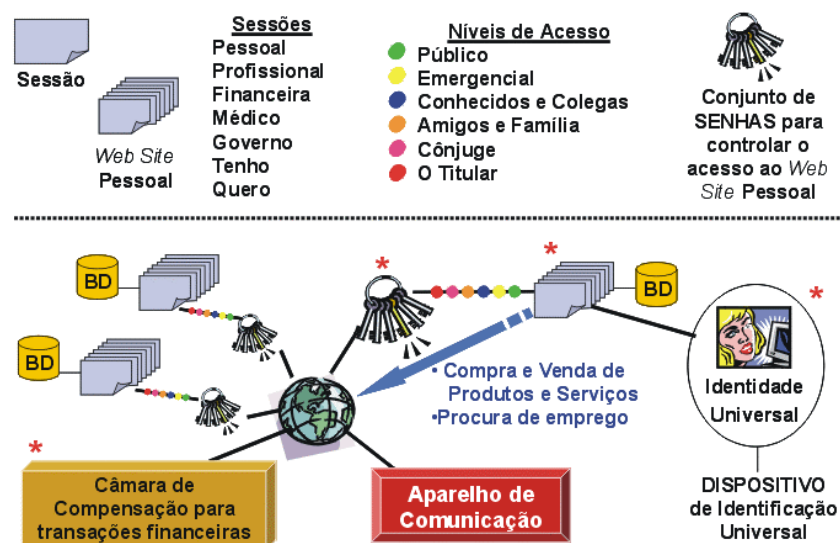


Figura 1.1: Modelo de Site Pessoal Universal proposto por David Siegel

Para que essa estrutura se torne viável, Siegel (ibidem) coloca a necessidade de quatro elementos fundamentais: a identidade universal, que identifica unicamente uma pessoa; um único endereço na Web (vinculado à identidade universal), onde cada cidadão pode organizar suas informações pessoais; um conjunto de senhas, facilitado por *softwares* sofisticados de administração de senhas para controlar o acesso ao *site* pessoal; e, uma câmara de compensação, com o objetivo de assegurar que todas as transações sejam legítimas, de forma que ao comprar algo *on-line*, basta fornecer a identidade universal e digitar uma senha para que o dinheiro eletrônico seja transferido.

David Siegel, também idealiza um dispositivo muito pequeno contendo a identidade universal que as pessoas podem ter consigo de várias maneiras: num relógio ou jóias, embaixo da pele, em cápsulas ingeridas diariamente, entre outras. Todos os lugares de qualquer tipo de comércio estão sempre conectados à Internet e possuem equipamentos de detecção deste dispositivo. Com isso, não há necessidade de digitar a identidade universal sempre que precisar dela.

Outro equipamento previsto para 2010 é o Aparelho de Comunicação (AC) que possui contínua conexão à Internet via satélite. Ele é movido à bateria e funciona como telefone, computador, relógio, identidade universal e televisão onde as funções são executadas sob comando de voz. O AC pode ser adaptado a diversos equipamentos de uso diário, como relógio, carro, câmeras de vídeo, etc, e pode ser configurado para enviar todo tipo de informação para o *Web Site* Pessoal de forma instantânea.

## **1.2 Objetivos**

### **1.2.1 Objetivo Geral**

Com base na idealização de David Siegel (SIEGEL, 2000), descrita anteriormente, propõe-se neste trabalho modelar uma estrutura de Portal Pessoal Universal que centraliza o acesso a *Sites* Pessoais Universais.

### **1.2.2 Objetivos Específicos**

Construir o modelo sobre uma estrutura flexível que permita a inclusão de novas sessões.

Validar o modelo através da construção de sessões predefinidas que possam ser utilizadas como referência na elaboração do *Site* Pessoal.

Identificar e propor soluções tecnológicas existentes na atualidade e que sejam viáveis para a materialização de algumas das atividades futuristas apresentadas.

### **1.3 Justificativa**

Considera-se que uma das contribuições deste trabalho é colaborar para colocar o poder da informação e do conhecimento a serviço da melhoria das condições de vida do povo brasileiro, apresentando aplicações que realmente agreguem valor para o indivíduo. Propõe-se aqui, uma maneira de incentivar a utilização da Internet, de forma a contribuir para a inclusão do maior número possível de cidadãos na Sociedade da Informação, expandindo rapidamente uma *cultura digital popular* pelo país. Isto ao mesmo tempo, contribui para que o povo brasileiro tenha condições de competir no mercado global. A idéia é oferecer para cada cidadão um espaço próprio na *Web*, para que possa armazenar e organizar todo tipo de informação a seu respeito com efeitos legais, a fim de possibilitar a recuperação das mesmas em qualquer tempo e lugar.

### **1.4 Limitações**

Dada a característica de flexibilidade do modelo proposto, optou-se por não apresentar todas as sessões propostas por Siegel, uma vez que se considera a possibilidade de criação de sessões com características diferentes.

Faz-se necessário deixar claro que garantir segurança numa estrutura com níveis de acesso é um trabalho significativamente complexo. Dessa forma, a segurança do *Site* Pessoal proposto está colocada na forma de sugestão e, portanto, não é discutida na sua totalidade.

## 1.5 Estrutura do Trabalho

Este trabalho está organizado de forma que facilita a leitura e a compreensão desta aplicação, apresentando no capítulo 2 aspectos que precisam ser considerados ao se desenvolver *sites* para a Internet, onde é realizada a revisão literária das tecnologias envolvidas nesta dissertação. No capítulo 3, está a descrição em detalhes de um modelo de *Site* Pessoal que é a essência deste documento. O capítulo 4 expõe o Portal Pessoal Universal, detalhando o modelo de um *software*, o Construtor de *Sites* Pessoais Universais (CSPU), sugerido para auxiliar na construção de um *Site* Pessoal através de perguntas dirigidas ao seu utilizador. Já as considerações finais que relatam as conclusões desse trabalho e determinados aspectos do sistema que serão desenvolvidos e/ou validados no futuro estão explanados no capítulo 5. As bibliografias que foram utilizadas como referência para o desenvolvimento desta dissertação, bem como os *sites* que foram consultados para ampliar o conhecimento no assunto do ponto de vista prático, estão relatados no capítulo 6.

## **2 ASPECTOS DE DESENVOLVIMENTO DE UM SITE**

O desenvolvimento de um *site* de sucesso deve seguir uma premissa básica: primeiro o planejamento, depois a construção propriamente dita. Seguindo esse processo, torna-se mais provável que tanto o conteúdo (o produto) quanto a maneira com que ele é apresentado (o serviço) sejam de qualidade. Um *site* também é um projeto visual que quando executado com seriedade e profissionalismo repassa credibilidade e confiança aos seus visitantes. Não é necessário fazer uso de todos os recursos disponíveis no mercado para que o *site* seja atrativo, essa atitude pode torná-lo pesado e, portanto, demorado para carregar (TORRES, COZER, 2000). Aliar simplicidade e eficiência muitas vezes é suficiente para conquistar a satisfação e fidelidade do internauta.

É interessante que se saiba que cada imagem de um *site* é um arquivo separado, assim como scripts, som, vídeo digital ou mini-aplicativo Java, que também são carregados com a página. O que se denota é que existem muitos materiais envolvidos que precisam de análise e organização específica para facilitar a construção e posterior manutenção (LOWERY, 2001).

Sendo assim, na seqüência estão descritos alguns aspectos importantes a serem considerados no desenvolvimento de um *site*, como planejamento, construção, segurança, organização e armazenamento de informações, além da modelagem lógica dos dados.

### **2.1 Planejamento**

Um bom começo é a definição de metas e objetivos, pois é muito difícil ter motivação para fazer algo sem saber onde se quer chegar. Metas estão relacionadas com áreas funcionais e devem ser encaradas visando um objetivo maior tendo um curto prazo definido para sua conclusão (OLIVEIRA, 2000), ou seja, são etapas estabelecidas, que ao serem alcançadas estão realizando parte de alguma coisa maior (TORRES,

COZER, 2000). Assim, torna-se possível atingir o objetivo como um todo, sem correr o risco de jamais transformá-lo em realidade.

Lowery (LOWERY, 2001) propõe que outras idéias – como mensagem, público, os vários aspectos do orçamento e o tempo necessário para a visualização da página na *Web* – sejam avaliadas. Através da definição da mensagem, esclarece-se o que se quer dizer através do *site*. O fato de identificar o público e seus principais hábitos, ou seja, quem visitará o *site* e o que espera encontrar nele, torna mais fácil definir toda a comunicação visual, além de elaborar o plano de *marketing* do negócio em questão. Saber quanto se pode investir também é um fator muito importante, à vista disso a equipe de desenvolvimento tem como destacar soluções viáveis dentro do orçamento que possui. Finalmente, deve ser considerado com seriedade o tempo de carga da página, considerando a velocidade dos *modems* mais comumente usados, já que, de acordo com Torres (TORRES, COZER, 2000), a maioria (33%) das pessoas espera somente até 15 segundos pela apresentação da página e apenas 5% dos internautas aguardam o tempo que for preciso. Por isso, é importantíssimo que as imagens sejam muito bem elaboradas, pois são os arquivos mais pesados e de maior atrativo, para que a página seja carregada antes do tempo limite para a maior parte dos internautas.

Se uma empresa tiver o objetivo de utilizar o *site* apenas como uma ferramenta de *marketing*, é necessário estabelecer um plano de negócios, que envolve planejamento operacional, de *marketing*, de crescimento, de retorno, e outros, que fazem parte do planejamento estratégico e é refletido no plano financeiro, o que possibilita fazer previsões de investimentos e captação desses recursos. Esse plano pode ser realizado com o auxílio de um breve documento que reflita a empresa ou negócio em questão, elaborado com respostas a perguntas básicas como: “Qual é o negócio? Qual é o produto? Qual é o funcionamento do negócio? Qual é o público-alvo? O que quero dos visitantes? Porque o negócio está sendo montado? Quais são os objetivos? Quais são as metas para que os objetivos sejam alcançados? Quanto tempo é necessário para desenvolver o trabalho por completo?” (TORRES, COZER, 2000).

Essa atitude, segundo Torres (*ibidem*), permite que sejam definidas questões referentes à organização interna da empresa – em todos os níveis, do raciocínio e das propostas da diretoria da empresa – e ao controle, onde são identificadas formas de

avaliação e monitoração da produção interna. A integração entre todas as pessoas que tem algum tipo de envolvimento com a empresa, incluindo diretoria, funcionários, clientes, fornecedores, colaboradores, enfim, todos trabalhando com objetivos comuns em mente e motivados para realizá-lo, também precisa de atenção especial. Por último, é imprescindível a definição dos meios de retorno utilizando instrumentos e formas de captação de recursos financeiros, humanos ou qualquer outro que seja necessário.

Mas, se o objetivo for utilizar o *web site* como uma ferramenta de estratégia da empresa, segundo David Siegel (SIEGEL, 2000), é necessário estabelecer uma mudança profunda na maneira como um *site* é visualizado. Se ao invés de apenas apresentar produtos ou serviços, o *site* oferecer o que os clientes esperam encontrar, certamente esse artifício passa a ser um importante aliado para colocar a empresa ou negócio em definitivo no mundo da Internet.

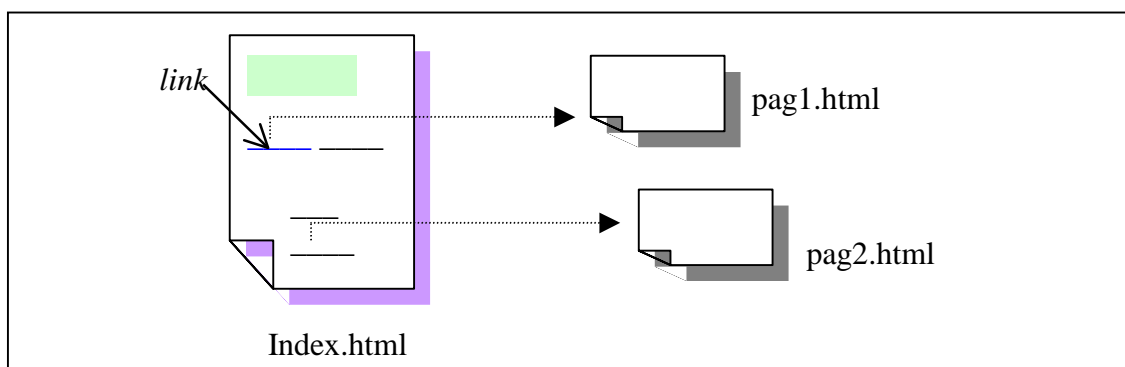
É importante saber quem são os “clientes eletrônicos”, ou seja, conhecer o que esperam encontrar num *site*. Colocar clientes e funcionários em contato ajuda nesse processo, promove uma conversação para que as demandas solicitadas sejam atendidas mais precisamente, e faz com que se sintam mais seguros (da mesma forma que se sentiriam no caso de dirigirem-se pessoalmente à empresa). Dessa forma, estabelece-se um canal de comunicação entre as partes e a empresa ganha a fidelidade de tal cliente a cada nova transação realizada. A equipe de desenvolvimento do *site* pode facilitar o diálogo não só entre clientes e funcionários, mas apresentar maneiras de um cliente conversar com outros clientes da mesma empresa, como em salas de “bate-papo” no próprio *site*. Assim, podem ter idéia de como será seu atendimento através da experiência de outras pessoas. Afinal, se os clientes estiverem felizes em seu relacionamento com uma empresa, certamente voltarão e farão propaganda dos serviços ou produtos utilizados.

Uma vez que se sabe quem são os clientes, o próximo passo é dividi-los em grupos, estabelecendo prioridades de dedicação aos mesmos. Daquele que receber mais alta prioridade, destaca-se um representante fictício e imagina-se, com base no perfil elaborado, tudo o que ele deseja encontrar, desde a maneira com que esse personagem chega ao *site* até o momento em que coloca o *site* em questão na lista de seus favoritos. O mesmo processo é repetido para os demais grupos destacados.

No mais, é executar o trabalho fazendo orçamentos, documentando todo o desenvolvimento, contratando o pessoal necessário para a codificação do *site* em si, a fim de colocar em prática tudo o que foi planejado.

## 2.2 Construção

Um *Web Site* é uma coleção de páginas hipermídia interligadas através de palavras-chave específicas que são conhecidas como *links*, como mostra a Fig. 2.1, onde a primeira página é denominada *home page*. Nessas páginas é possível utilizar recursos de imagem, vídeo, som, animação e textos, é claro.



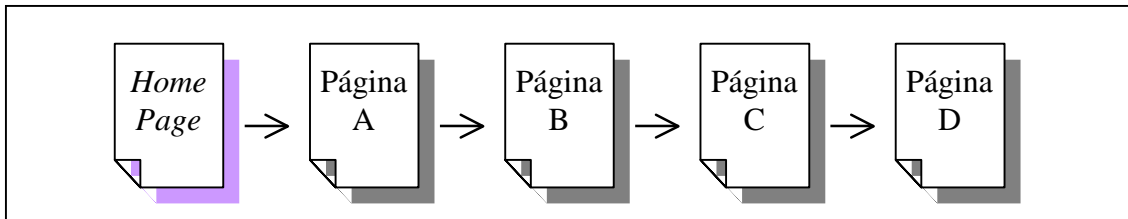
**Figura 2.1: Exemplo de hipertexto**

Para organizar os *links*, geralmente utiliza-se uma barra de navegação. Ela deve ser organizada de tal forma que o nome dado aos *links* refiram-se exatamente para onde o visitante será remetido. Também é interessante que seja possível ir de onde se está para qualquer outro lugar do *site*. Com isso, será possível encontrar com mais rapidez uma determinada informação. O modelo de navegação deve levar em conta o tempo de carregamento das páginas, porque a demora da navegação implica em perda de tempo e dinheiro (ligação telefônica e provedoria) ao visitante (TORRES, COZER, 2000).

Segundo Lowery (LOWERY, 2001), existem vários modelos para estruturar a navegação de um *site*, cada um tem um propósito específico, mas podem ser combinados de acordo com a intenção do projetista. São eles: **linear**, **hierárquico**, **raio e centro**, e **web desinger completo**. No primeiro, as páginas são dispostas de maneira seqüencial, onde a *home page* é a página inicial de navegação e leva o visitante para a

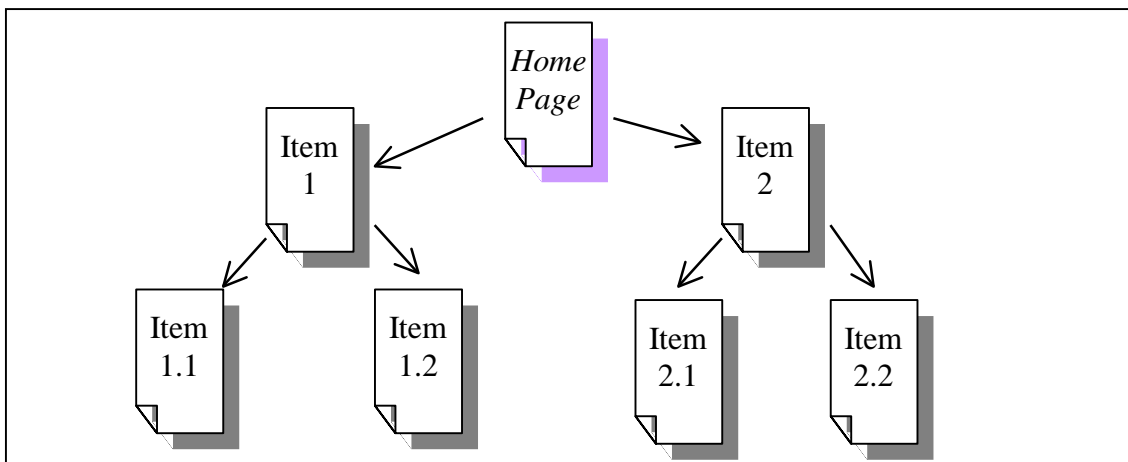


página A, essa, por sua vez, direciona o internauta à página B, que o remete à página C e assim sucessivamente, como mostra a Fig. 2.2. Este estilo é mais adequado para sistemas de treinamento por computador ou para outros onde se quer total controle dos atos do usuário.



**Figura 2.2: Modelo linear de navegação**

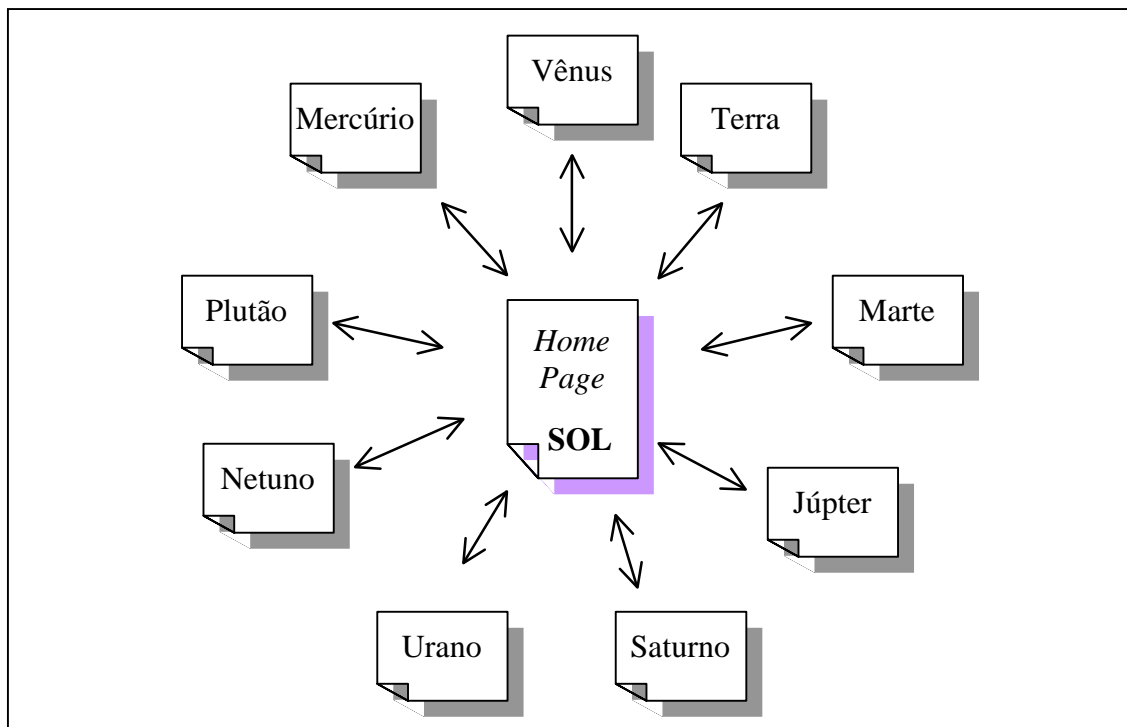
Há também o **estilo hierárquico**, que lembra um organograma, no qual a idéia principal é elaborada na *home page*, e desta partem os itens que, por sua vez, são ramificados em sub-itens, o que lembra o formato de uma árvore, como pode ser observado na Fig. 2.3.



**Figura 2.3: Modelo hierárquico de navegação**

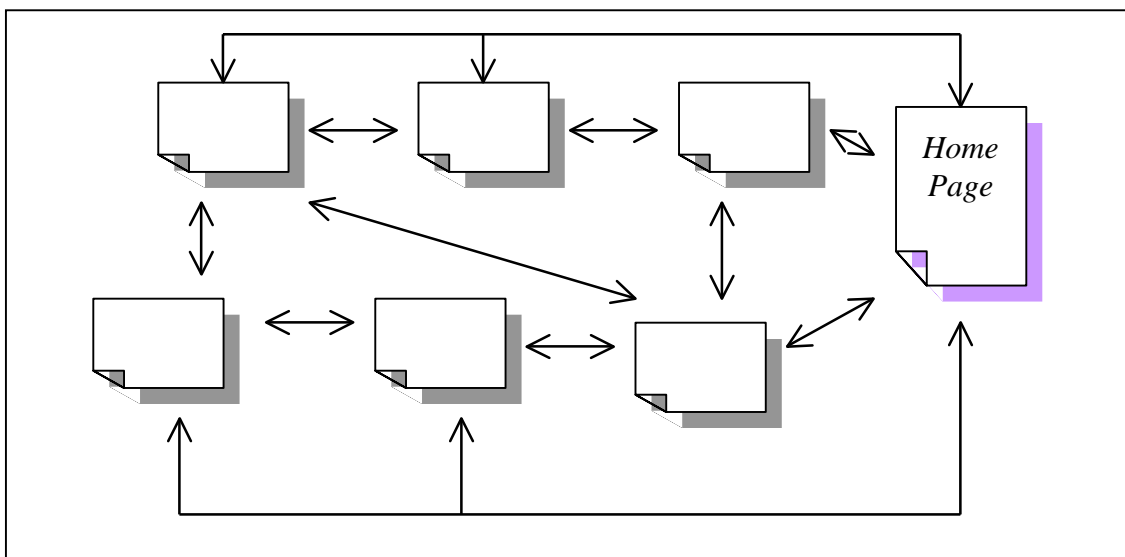
Há temas, onde o **modelo raio e centro** se aplica muito bem, como é possível ver no exemplo dos planetas em torno do sol mostrado na Fig. 2.4. Outro exemplo é de um zoológico, onde os animais não têm interação entre si, apenas com o tópico central. Nesta abordagem, não há relação entre os itens das ramificações. A vantagem é que em apenas poucos saltos o internauta navega por todo o *site*. O inconveniente está na necessidade de ter que sempre voltar à *home page* para visitar outra página do *site*. Por

esse motivo, muitos projetistas optam pelo uso de quadros (*frames*) disponibilizados durante toda a navegação, onde os principais itens são colocados para fornecer acesso mais rápido.



**Figura 2.4: Modelo raio e centro de navegação**

O método que mais tira vantagens das capacidades de *hyperlink* da *Web* é o chamado **web design completo** (Fig. 2.5). Neste, é oferecida a oportunidade do internauta visitar todas as páginas a partir de qualquer uma do *site*. Essa maneira pouco estruturada permite que o usuário navegue pelo *site* de acordo com sua necessidade, sem haver qualquer tipo de controle por parte do *designer*. Em algumas situações pode acontecer de o visitante sentir-se perdido por não encontrar um caminho pré-determinado, em função disso normalmente é colocado à disposição do internauta um mapa do *site*, onde é apresentada toda a estrutura de conexão entre as páginas.



**Figura 2.5: Modelo web designer completo de navegação**

O grande produto da Internet é a informação, e é ela que deve ser valorizada. O grande serviço da Internet é como a informação é veiculada, ou seja, o projeto de um *site* na Internet. Portanto, montar um *site* não é simplesmente ter uma idéia e programar o código das páginas, mas sim fazer um projeto bem elaborado, colocando-o no papel, simulando seu comportamento no dia-a-dia, verificando todas as áreas envolvidas – grupos de clientes, custos iniciais, de desenvolvimento, de hospedagem, projeção de custos de veiculação de propaganda, etc... – antes de colocá-lo no ar, como foi abordado no item 2.1. Também é muito importante pensar no produto em questão, ou seja, no conteúdo do *site* e em seu público alvo, além de como serão feitas as atualizações (TORRES, COZER, 2000).

Um *site* bem projetado é aquele que alia simplicidade e eficiência. É preciso ter cuidado ao utilizar muitos recursos tecnológicos: mais vale um *site* leve, rápido e objetivo que um pesado, que demore a carregar e que possua recursos dispensáveis. Enfatizando a simplicidade, os *sites* ficam menores, mais rápidos, menos confusos, mais atraentes, mais fáceis de atualizar ou expandir, enaltecem o produto e, conseqüentemente, conquistam muitos visitantes.

Embora o conteúdo e o projeto estejam bem fundamentados e estruturados, não se pode esquecer do *design*, ou seja, da aparência do *site*. Um ambiente com cores, imagens e textos apresentados de forma integrada com o que está sendo informado, se

torna atraente para o visitante, facilita o entendimento do assunto e o instiga a voltar sempre que precisar.

O *design* de uma página é, antes de tudo, uma peça de comunicação visual. E deve ser feita de acordo com o público-alvo e com o objetivo da página. É necessário que o produto (conteúdo) e o serviço (o que o *site* faz) em questão estejam bem definidos, que haja um único foco central para atrair a atenção do visitante (ao invés de vários), e uma frase curta definindo o que é o *site* para que ele saiba claramente do que trata aquele endereço virtual (TORRES, COZER, 2000).

Outro cuidado que se deve ter é com o uso de linguagens de programação que não são compatíveis com a maioria dos *browsers* do mercado e com o uso de recursos que só estão disponíveis através da instalação de um *plug-in*, isto é, uma extensão para que o *browser* consiga visualizar o recurso. É preciso fazer diversos testes e colocar avisos, por exemplo, a respeito do uso de *plug-ins* ou de uma resolução de vídeo mais adequada, para que o visitante não fique sem saber o que aconteceu quando algo não funcionar direito. É necessário pensar bastante antes de usar tais recursos ou versões específicas de *browsers* para ter acesso a um *site*. Uma boa alternativa é disponibilizar duas versões: uma para quem tem os requisitos necessários e outra para quem não os têm.

A respeito da composição dos textos para a Internet é interessante sempre utilizar a linguagem mais coloquial e direta possível. Dessa forma, cria-se uma relação de intimidade com o usuário, deixando-o mais à vontade num *site*. Os textos devem ser resumidos para serem entendidos em poucas palavras. O melhor é optar por parágrafos curtos, transformando algumas frases importantes em notas ou *links* do tipo “*Mais informações...*”, ricos em informações com detalhes, usando um vocabulário facilmente compreendido, além de dedicar atenção especial à gramática e ortografia. Desta forma, quem não tem um domínio completo do idioma aplicado ao *site*, terá menos dificuldade para entender o que está sendo apresentado (TORRES, COZER, 2000).

Segundo a *Global Reach*, existem 2,9 milhões de pessoas que falam português na Internet, ocupando o 11º lugar dentre as mais usadas na Internet. Por outro lado, são 128 milhões de pessoas que têm o inglês como língua nativa, ou seja, um público quase 45 vezes maior (TORRES, COZER, 2000). Diante desta estatística, é muito interessante

disponibilizar o *site* num segundo idioma, já que a idéia é torná-lo acessível para o maior número de pessoas possível, independente do idioma que tenham conhecimento.

Uma vez definidos o conteúdo, o projeto e o *design* do *site*, chega o momento de realizar a codificação, incluindo a criação e a edição adequada das imagens. Neste momento é importante lembrar-se de acrescentar comandos Meta no início do código HTML de cada página para facilitar a indexação por ferramentas de busca, tentar fazer com que as imagens sejam do menor tamanho possível sem perder sua qualidade, sempre que possível utilizar comandos HTML ao invés de imagens (no caso de *links* representados em botões) para que a página fique mais rápida, evitar usar recursos onde *plug-ins* sejam necessários, construir e testar as páginas para que sejam compatíveis com o maior número possível de *browsers* disponíveis no mercado e suas versões, fazer testes e adaptações com resoluções diferentes, além de cadastrar o *site* nas principais ferramentas de busca.

É importante destacar que a variedade de linguagens que codificam páginas para a Internet tem crescido rapidamente, como asp, jsp e xml, entre outras, que atuam no sentido de apresentar um *design* atraente, mas sem muito custo para ser visualizada. A tendência é que estas melhorem ou ofereçam recursos cada vez mais interessantes e que novas linguagens sejam colocadas nesse mercado. Portanto, este trabalho se limita a propor o conteúdo e o projeto para um modelo genérico de *Site* Pessoal.

## 2.3 Segurança

Sabe-se que no mundo real não existem sistemas totalmente seguros e o mundo virtual não é diferente. Por maior que seja a proteção adotada, sempre haverá a possibilidade de invasões, roubos e ataques. Então, é importante conhecer os esquemas de segurança existentes, a fim de escolher qual o mais indicado para executar a proteção de informações. Afinal, a Internet tem sido utilizada para realizar diversos serviços do cotidiano, como compras, serviços bancários, investimentos, além de negócios ou troca de informações confidenciais, via *e-mail*.

### 2.3.1 Criptografia

Falar em segurança significa dizer que um documento não será lido, alterado ou obtido por alguém não autorizado. Para que isso aconteça é preciso que as informações sejam embaralhadas ou criptografadas, sendo compreensíveis apenas para as pessoas que sabem como decifrá-las. Esse processo é conhecido como criptografia (PASQUAL, 2001).

Os métodos de criptografia utilizam “chaves” para cifrar e decifrar uma mensagem. Esta chave é uma seqüência de caracteres convertidos em um número, medido em bits – quanto maior o tamanho da chave, mais caracteres devem ser utilizados para criá-la e, conseqüentemente, maior é a dificuldade para ser descoberta. Existem duas maneiras básicas de fazer a codificação e decodificação de informações, são elas (IGNACZAK, 2002):

- A criptografia de chave única, ou simétrica; e,
- A criptografia de chaves pública e privada, ou assimétrica.

Na criptografia simétrica, ou de chave única, a chave que cifra as mensagens é a mesma que decifra. Neste caso, num grupo de trabalho ou de amigos a chave é de conhecimento de todos. Métodos desse tipo são considerados rápidos e difíceis de decifrar com chaves não menores que 128 bits. O problema é que se o grupo for ficando grande, a chave passará a ser considerada pública e a segurança da troca de mensagens ficará ameaçada. Logo, estes métodos têm mais utilidade para codificar arquivos que serão acessados por uma única pessoa, do que para corresponder-se com amigos.

Já a criptografia assimétrica, ou de chaves pública e privada, é um método onde um par de chaves é utilizado para criptografar e decriptografar mensagens eletrônicas que tenham sido enviadas através de caminhos desprotegidos. As duas chaves que compõe o par mantêm entre si uma relação matemática muito específica, pois quando uma chave cifra uma mensagem eletrônica, apenas a chave correspondente a ela pode complementar esta função, decifrando a mensagem e, portanto, autenticando seu remetente e sua integridade ao mesmo tempo. Conhecer uma das chaves não requer,

nem divulga o conhecimento da outra e, também, não se sabe qual delas é mantida em segredo por seu dono.

### 2.3.2 Função *Hash*

É um algoritmo que gera matematicamente o resumo de uma mensagem, também conhecido por valor *hash* que é único, pois utiliza o conteúdo da mesma, de forma que uma mínima alteração na mensagem original, resulta num resumo completamente diferente. Além de apresentar-se como a “impressão digital” de uma mensagem, o valor *hash* sempre tem o mesmo tamanho fixo, independentemente do tamanho da mensagem original (NOTOYA, 2002). As funções *hash* mais utilizadas são o MD5, que produz um resumo com tamanho de 128 bits e o SHA1 (*Secure Hash Algorithm*), que resulta num valor *hash* com 160 bits de tamanho. Isto significa que a probabilidade de serem gerados resumos iguais de mensagens diferentes usando o padrão MD5 é da ordem de  $2^{128}$  e pelo padrão SHA1 é da ordem de  $2^{160}$  (PASQUAL, 2001).

Para uma função *hash* ser considerada segura é preciso que os seguintes aspectos sejam garantidos (ROCHA, 2002) (NOTOYA, 2002):

- O tamanho da mensagem original é irrelevante;
- O tamanho do resumo gerado a partir da aplicação do algoritmo é sempre o mesmo;
- A função *hash* deve ser de fácil aplicação;
- Deve ser computacionalmente impossível gerar a mensagem original a partir do resumo da mesma – direção única;
- Deve ser impraticável o fato de duas mensagens diferentes resultarem o mesmo valor *hash* – forte resistência à colisão;
- Não pode haver igualdade no resumo de um par de mensagens que façam sentido – fraca resistência à colisão;

### 2.3.3 Assinatura Digital

A assinatura digital está diretamente relacionada com a aplicação da técnica de criptografia de chave pública. Não há uma ordem fixa na utilização das chaves, ou seja, a chave pública pode servir tanto para codificar quanto para decodificar uma mensagem. Quando é a chave privada que codifica as informações, o processo fica caracterizado como **assinatura digital**, provendo ao documento eletrônico confidencialidade (garante que pessoas não autorizadas tiveram acesso à mensagem), autenticidade (sabe se o remetente é realmente quem diz ser), integridade (constata que a mensagem não foi alterada após ter sido assinada), e não-recusa (após ter assinado a mensagem, o remetente não pode negar tê-lo feito) (BORTOLI, 2002).

O processo que caracteriza a verificação da integridade de uma mensagem assinada digitalmente acontece da seguinte maneira (Fig. 2.6): o resumo de uma mensagem é obtido através da aplicação da Função *hash* à mesma; este resumo de mensagem é codificado com a chave privada do remetente, o João; o resumo codificado é anexado à mensagem e esse bloco de informações é enviado para Ana; quando Ana, a destinatária, recebe o bloco de informações, reserva o resumo enviado com a mensagem e decifra-o com a chave pública de João; em paralelo, Ana aplica a Função *hash* à mensagem recebida; por fim, os dois resumos são comparados e ao verificar a igualdade dos mesmos Ana tem certeza que a mensagem assinada por João não foi alterada.

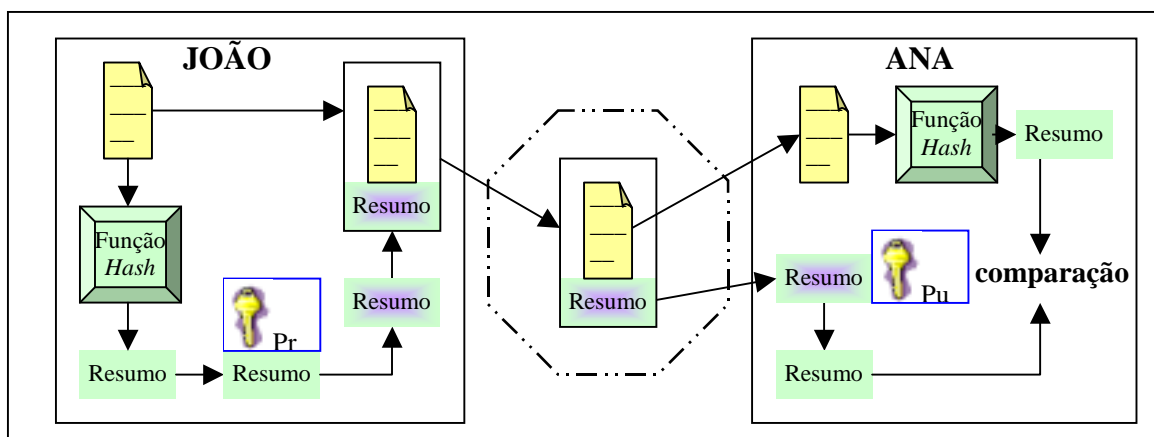


Figura 2.6: Assinatura Digital



A assinatura digital é gerada em função do conteúdo de um documento, com o auxílio das técnicas de criptografia assimétrica e função *hash*, mas necessita de outra informação que realize o vínculo entre ela e uma pessoa. Esse vínculo é caracterizado como autenticação que é classificado em 6 níveis (PASQUAL, 2001):

- Algo que se sabe – uma senha, por exemplo;
- Algo que se tem – um cartão magnético, por exemplo;
- Algo que se é – biometria, impressão digital ou leitura da íris, por exemplo;
- Onde se está – somente em determinado estabelecimento;
- Em que horário se está – somente em determinado horário;
- Com uma testemunha – uma pessoa que estava presente no ato da assinatura.

Para aumentar a segurança no ato da autenticação, é possível utilizar dois ou mais níveis dos apresentados.

### 2.3.4 Certificado Digital

O problema da utilização da assinatura digital é que qualquer pessoa pode usar uma assinatura. Então, como saber se a assinatura gerada sob o nome de uma determinada pessoa realmente pertence a ela?

É para resolver esta questão que surge o **Certificado Digital**, onde a assinatura digital tem o aval de uma Autoridade Certificadora, cujo objetivo é realizar um vínculo único entre pessoas e assinaturas digitais de forma confiável. Atualmente, o certificado digital é a base para a utilização da *Web* de forma segura. Empresas que vendem produtos pela Internet, bancos e outras entidades que necessitam de segurança, precisam adquirir um certificado digital de uma autoridade certificadora para prover acesso seguro.

O certificado digital contém, no mínimo, as seguintes informações (PASQUAL, 2001):

- A chave pública;

- O nome do usuário e endereço de *e-mail*;
- Data de validade da chave pública;
- Número de série do certificado digital;
- Nome da autoridade certificadora que emitiu o certificado; e,
- A assinatura digital da certificadora.

O certificado ITU-T *x.509 v3* é o padrão mais conhecido e aceito para certificados de chave pública (PASQUAL, 2001), sendo amplamente utilizado por muitos protocolos modernos de criptografia, inclusive o SSL (*Secure Socket Layer*). A versão 3 (v3) foi adotada pela indústria porque permite a inserção de dados arbitrários no certificado, que podem ser utilizados para propósitos variados.

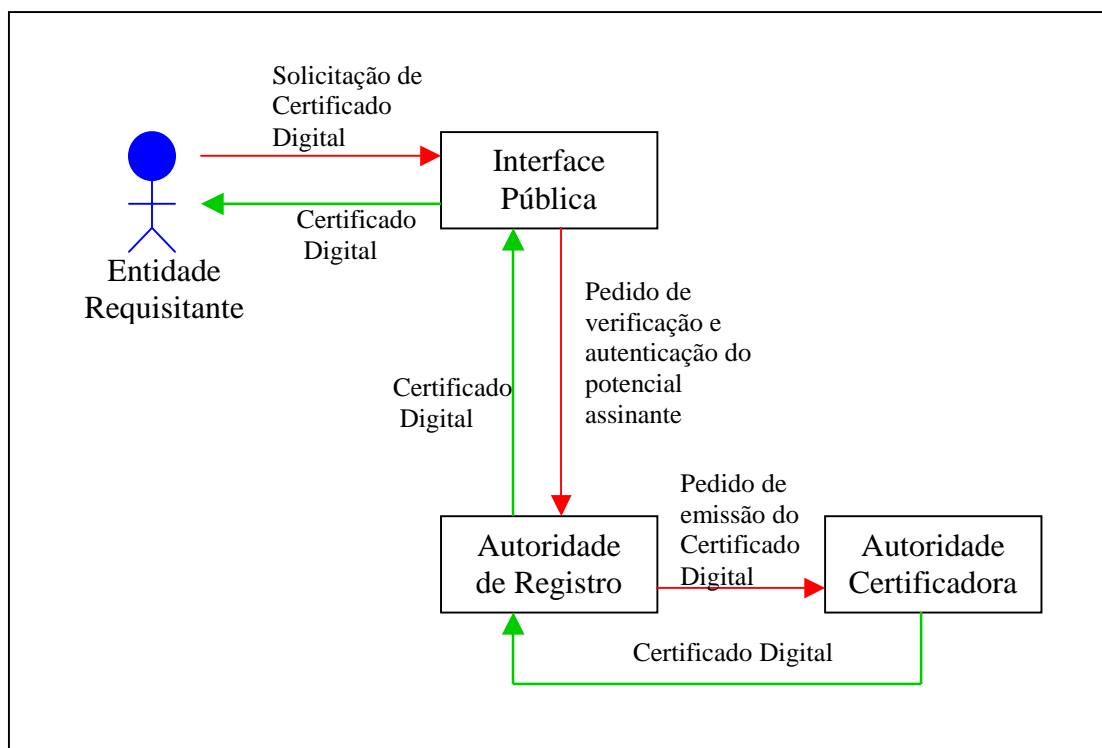
A autenticação descreve o processo de confirmação de identidade que o computador faz. Não se limita aos *sites* da *Web*, também é utilizada em aplicativos de rede onde o processo é semelhante. Quando se acessa um *site* da *Web*, o navegador apresenta o Certificado Digital do internauta. O *site* visualiza a informação no certificado digital e determina o que essa pessoa tem permissão para fazer. Dependendo do navegador, será necessário confirmar que se deseja apresentar o certificado. Geralmente, a senha do Certificado será solicitada numa caixa de diálogo e, em seguida o navegador envia o certificado digital ao *site* da *Web*. Quando o certificado é recebido, a data de validade é verificada e também a autoridade certificadora que o emitiu. Caso o *site* não confie na autoridade certificadora, o acesso ao *site* pode ser recusado.

A data de validade é adotada por questões de proteção contra ataques de força bruta. Todo navegador ou aplicativo de *e-mail* verifica essa informação para garantir que a data de recebimento do Certificado Digital, ou seja, das informações que ele está protegendo, está dentro do período de validade. Isso significa que quando a própria senha vence, tudo o que foi assinado com ela não será mais válido, a partir daquela data. Uma vez vencido, o Certificado Digital necessitará de renovação que é feita através do Centro de Certificação Digital.

### 2.3.5 Infra-estrutura de chaves públicas - ICP

É um conjunto de serviços responsáveis pela emissão de Certificados Digitais com base na técnica de criptografia de chave pública. Existem basicamente quatro entidades envolvidas nessa atividade: a entidade requisitante, uma interface pública, a Autoridade de Registro (AR) e a Autoridade de Certificação (AC).

O processo de distribuir chaves públicas de forma precisa e confiável àqueles que precisam codificar mensagens ou verificar assinaturas digitais faz uso de Certificados Digitais emitidos por uma Autoridade Certificadora para os usuários registrados nela. A emissão do certificado exige autenticação do usuário, realizada pela Autoridade de Registro (Fig. 2.7). Também fazem parte do escopo da ICP os serviços de renovação e revogação de certificados, verificação de status e confecção de cópia de segurança/recuperação da chave privativa do usuário (ROCHA, 2001).



**Figura 2.7: Processo de Emissão de Certificado Digital**

### 2.3.6 Protocolização de documentos eletrônicos

A protocolização digital tem o objetivo de associar data e hora confiáveis a um documento de forma que o valor jurídico possa ser garantido por longo período de tempo (NOTOYA, 2002). Isto significa que o documento protocolado tem sua existência confirmada a partir da data e hora conferidas a ele.

Documentos, de um modo geral, são a representação de um fato e normalmente estão na forma escrita sobre um papel. Porém, na atualidade também se passou a considerar outros meios de representatividade, tais como: texto, imagem, vídeo ou som digitalizados caracterizando o documento eletrônico, cujo registro legal consta na Medida Provisória 2200-2 que se encontra no anexo 2. Em outras palavras, Notoya (NOTOYA, 2002) cita que um documento “pode se constituir de uma seqüência de bits que pode ser interpretada por meio de *softwares* que possibilitam a verificação da expressão do pensamento ou vontade daquele que o formulou”.

No entanto, documentos eletrônicos precisam ter alguns atributos para adquirirem valor jurídico, como autoria, integridade, referência de tempo e não-repúdio (NOTOYA, 2002). Os dois primeiros aspectos são garantidos por meio da assinatura digital, pois é gerada de forma única com base no conteúdo do mesmo, onde a mínima alteração no documento invalida a assinatura. A questão tempo é relacionada à confiabilidade conferida à Autoridade de Datação, que precisa garantir determinados aspectos para que data e hora confiáveis sejam disponibilizadas e que não haja meios de serem alteradas depois de vinculadas a um determinado documento. Este fator é imprescindível no caso de disputas. Por último, o atributo não-repúdio se refere ao fato de que o autor de um documento não pode negar tê-lo feito após sua assinatura constar no mesmo. Isto é garantido por meio do Certificado Digital ao qual a assinatura está estritamente relacionada, pois ele é emitido sob o aval de uma Autoridade Certificadora, que já possui a verificação e autenticação da identidade do proprietário da assinatura digital.

O processo de protocolização de um documento eletrônico, de uma maneira geral, acontece da seguinte forma: o cliente gera um resumo do documento a ser protocolado, utilizando sua assinatura digital. Este resumo é enviado para a Autoridade de Datação que anexa data e hora confiáveis ao mesmo, assina e envia para o cliente o recibo de

datação (PASQUAL, 2001). Transmitindo o resumo do documento ao invés do próprio, garante a integridade e confiabilidade do mesmo, pois é impossível obter um documento através de seu resumo.

## **2.4 Organização e Armazenamento das Informações**

Atualmente, verifica-se que novas aplicações para sistemas de banco de dados estão sendo limitadas por restrições impostas pelo modelo relacional de dados. Dentre elas, pode-se citar aplicações CAD, CASE, banco de dados multimídia e banco de dados hipertexto.

Sendo assim, várias propostas têm sido apresentadas para tratar e suprir esses novos domínios de aplicação, entre elas destacam-se os bancos de dados orientados a objetos, seguindo o paradigma de orientação a objetos, e os bancos de dados relacionais-objetos, que estendem o modelo relacional de dados fornecendo um tipo de sistema mais rico, incluindo orientação a objetos e adicionando estruturas às linguagens de consulta relacionais, como a SQL, para tratar novos tipos de dados.

Os modelos relacional e entidade-relacionamento não suportam mais as necessidades dos dados das mais recentes aplicações. Além disso, freqüentemente aplicações tradicionais têm tido a necessidade de tratar dados de imagens e bancos de dados hipertexto, em função do uso difundido da Internet (SILBERSCHATZ, KORTH, SUDARSHAN, 1999).

Portanto, na seqüência serão apresentados conceitos relacionados ao paradigma de orientação a objetos e ao paradigma relacional-objeto para demonstrar como essas novas maneiras de modelagem podem auxiliar na organização e interação dos dados que as mais recentes aplicações têm apresentado.

### **2.4.1 O Modelo de dados Orientado a Objetos**

Os sistemas orientados a objetos podem representar melhor o mundo real, considerando que a forma de percepção e raciocínio humano está intimamente ligada ao

conceito de objetos. Assim, a modelagem de sistemas é realizada de maneira natural e reflete fielmente situações da realidade. Na seqüência são abordados os aspectos que caracterizam esse modelo.

#### **2.4.1.1 A estrutura objeto**

Um objeto geralmente tem associado a ele um conjunto de variáveis que contém os dados, atributos ou características do mesmo e um conjunto de métodos Internet (SILBERSCHATZ, KORTH, SUDARSHAN, 1999).

Os atributos de um objeto são características individuais que o diferenciam dos demais. Podem incluir informações sobre seu estado e são expressos como uma variável (LEMAY, 1998). A variável é usada para armazenar o valor do atributo e um método é usado para ler e atualizar o seu valor (SILBERSCHATZ, KORTH, SUDARSHAN, 1999).

#### **2.4.1.2 As classes**

Uma classe é um modelo que centraliza objetos com características semelhantes (LEMAY, 1998). Normalmente, em um banco de dados existem diversos objetos que possuem atributos e comportamentos semelhantes, portanto eles são agrupados em classes. Os objetos são então chamados de instâncias da classe a qual pertencem, diferenciando-se entre si apenas pelo valor relacionado às suas características.

#### **2.4.1.3 A herança**

Um banco de dados orientado a objetos normalmente utiliza uma grande quantidade de classes, sendo que muitas delas apresentam características semelhantes. Assim, para evitar a repetição de tais particularidades, opta-se por organizar essas classes num esquema de hierarquia de especialização. As especializações de uma classe são tidas como subclasses da classe superior. Para obter uma visão mais clara, tomamos um exemplo, onde a classe Veículos é a principal, e as subclasses Automóveis e *Pick-ups*, como suas especializações. A assim chamada superclasse contém os atributos referentes a veículos, como motor, cor, combustível, etc, e as subclasses descrevem

características específicas de seu domínio, no caso de *Pick-ups* pode-se citar o tamanho da caçamba, particularidade não encontrada na subclasse Automóveis, e assim por diante.

O que se observa é que não há necessidade de reescrever variáveis como motor, cor, combustível, entre outras, pois elas são herdadas da superclasse. Além dos atributos, os métodos descritos na superclasse também são herdados. É nessa facilidade de organização das informações na modelagem de uma situação real, sem haver repetição de variáveis e métodos (podendo, inclusive, serem reutilizados sempre que for necessário), que está a grande vantagem da aplicação da herança.

#### **2.4.1.4 Identidade**

Na medida em que se destaca um representante de uma determinada classe obtém-se um objeto. De outra forma, pode-se dizer que objeto é uma instância de uma classe, onde às variáveis são atribuídos valores quaisquer. Desta maneira, esse representante ou objeto precisa ser diferenciado de outras instâncias dessa classe, já que os valores das variáveis podem ser iguais exceto um, aquele que vai identificá-lo unicamente.

No mundo real, uma pessoa é identificada com precisão, através de um conjunto de características, como nome completo, número de identidade, CPF, endereço. No mundo dos objetos, utiliza-se um identificador pelo qual se fará a localização exata de um determinado objeto. Este identificador pode ser efetivado por um Valor, utilizado em sistemas relacionais, onde existe a figura da chave primária identificando unicamente uma tupla; um Nome, que pode ser fornecido pelo usuário, utilizado em sistemas convencionais, como exemplo, nomes de arquivos independente de seu conteúdo; ou na forma Embutida, onde não é necessária a intervenção do usuário, é verificado em sistemas orientados a objeto onde um identificador único é associado automaticamente pelo sistema no momento da criação de cada instância.

#### **2.4.1.5 Linguagens orientadas a objetos**

Os conceitos vistos até o momento, precisam ser expressos em alguma linguagem para tornarem-se praticáveis. Essa expressão pode ser feita através de um banco de

dados relacional puro, onde os conceitos de orientação a objeto são utilizados apenas como uma ferramenta de projeto e os objetos são convertidos em um conjunto de relações.

Outra maneira é fazer uso de uma linguagem que é usada para manipular banco de dados, o que amplia o leque de escolha em que os conceitos podem ser integrados. A primeira escolha é estender uma linguagem de manipulação de dados como a SQL, pela adição de tipos complexos e orientação a objetos, sendo conhecidos como sistemas relacionais-objeto. Uma segunda opção é escolher uma linguagem de programação orientada a objeto e estendê-la para tratar bancos de dados, onde são chamadas de linguagens de programação persistentes.

Há diferentes situações, com requisitos e prioridades específicos, onde uma das formas descritas acima será mais indicada. Por isso, é importante conhecer o comportamento, as vantagens e desvantagens de cada uma para que a escolha final seja acertada.

#### **2.4.1.6 Linguagens de programação persistentes**

A diferença entre linguagens de bancos de dados e as tradicionais está no fato de que as primeiras manipulam diretamente os dados que são persistentes, isto é, aqueles que continuam a existir mesmo após o fim da execução da aplicação. Uma relação e suas tuplas em um banco de dados são exemplos de dados persistentes, já as linguagens tradicionais têm como únicos dados persistentes os arquivos que elas manipulam diretamente.

O que se verifica é que o acesso ao banco de dados é apenas uma parte de uma aplicação, já que a interface com o usuário e outros cálculos que sejam necessários ficam por conta de uma linguagem de programação tradicional. Por outro lado, as linguagens tradicionais não têm como fazer acesso a banco de dados senão utilizando alguma outra que o faça, como a SQL, por exemplo.

Uma linguagem de programação persistente é uma linguagem de programação estendida com estrutura para tratar dados persistentes.



Ela pode ser diferenciada de linguagens com SQL embutida na medida em que os tipos de dados são diferentes, já que as linguagens *host* e de manipulação de dados o são, ficando sob a responsabilidade do programador qualquer tipo de conversão necessária. Além disso, há a questão da conversão do formato orientado a objeto da linguagem *host* e o formato relacional das tuplas no banco de dados, acarretando grande quantidade de código para que a tradução seja feita. Outro, porém, fica por conta da necessidade de se escrever explicitamente o código que faz a leitura e atualização no banco de dados.

Estas questões não apresentam problema algum numa linguagem de programação persistente, pois os objetos podem ser armazenados no banco de dados sem qualquer tipo explícito ou mudanças de formato (que se forem necessárias, são feitas de forma transparente) e a manipulação dos dados é realizada sem a necessidade de se escrever claramente os códigos que realizam essa tarefa.

Entretanto, existem alguns inconvenientes nas linguagens de programação como o poder de sua estrutura, que torna relativamente fácil a ocorrência de erros de programação que danifiquem o banco de dados, e a complexidade da linguagem faz com que a otimização de alto nível, como a redução de I/O em disco, seja mais difícil de ser efetivada.

#### **2.4.1.7 Persistência de objetos**

Para fazer com que os objetos persistam ao final da execução de uma aplicação de banco de dados é possível aplicar um dos seguintes métodos (SILBERSCHATZ, KORTH, SUDARCHAN, 1999):

- Persistência por classe: a forma mais simples, mas menos indicada, já que desse modo todas instâncias são definidas como objetos persistentes e em muitas situações podem ocorrer onde é conveniente se ter objetos persistentes e outros transientes;
- Persistência por criação: quando o objeto é criado, o argumento que indica se ele é persistente ou transiente é acrescentado na sintaxe do comando de criação;

- Persistência por marcação: todos os objetos são criados como transientes e aqueles que serão persistentes devem ser claramente marcados antes do término da aplicação;
- Persistência por referência: um determinado objeto é declarado explicitamente como persistente, sendo então, considerado objeto raiz. Todos os demais que fizerem algum tipo de referência ao mesmo e outros que fizerem referência a estes, e assim por diante, são automaticamente considerados como persistentes. A vantagem desse método é que passa a ser possível tornar uma estrutura de objetos que se inter-relacionam como persistentes através da declaração de persistência em somente um deles, o objeto raiz. Entretanto, o sistema de banco de dados tem responsabilidades e custos a serem considerados ao percorrer cadeias de referências para detectar quais objetos são persistentes.

#### **2.4.1.8 Identidade de Objetos e Ponteiros**

Quando um objeto persistente é criado ele recebe um identificador de objeto persistente. Mas, em linguagens de programação orientadas a objeto que não têm extensão para tratar persistência, a identidade retornada é transiente, perdendo o sentido após o término da aplicação.

Existe uma relação entre identidade de objetos e ponteiros utilizados em linguagens de programação, onde ponteiros para localizações físicas no armazenamento de informações podem ser uma maneira de representar a identidade embutida de objetos.

Uma vez que a associação de um objeto com uma localização física de armazenamento pode mudar com o tempo, pode-se considerar vários graus de persistência de identidade:

- Intraprocedimento: a identidade permanece válida somente dentro de um procedimento, assim como variáveis locais;

- Intraprograma: a identidade persiste apenas dentro de um programa, da mesma forma que variáveis globais ou ponteiros de memória principal ou virtual;
- Interprograma: a persistência da identidade é verificada a partir de uma execução de programa por outra. Um exemplo está nos ponteiros para dados do sistema de arquivo no disco que podem ser alterados se o modo pelo qual o dado é armazenado no sistema de arquivo é alterado;
- Persistência: a identidade de objetos continua valendo tanto entre as execuções de programas quanto entre as reorganizações estruturais dos dados, sendo esta a forma de persistência de identidade exigida para sistemas orientados a objetos.

Em linguagens que suportam a persistência, como C++, os identificadores de objetos persistentes são implementados como “ponteiros persistentes”, um tipo de ponteiro que permanece válido mesmo com o fim da execução de um programa e através de algumas formas de reorganização de dados. Ele pode ser usado da mesma maneira que ponteiros de memória e é possível entender um ponteiro persistente como um ponteiro para um objeto no banco de dados.

#### **2.4.1.9 Armazenamento e acesso a objetos persistentes**

Armazenar um objeto em um banco de dados implica em armazenar a parte de dados individualmente para cada objeto e o código que implementa os métodos também devem ser guardados como parte de um esquema de banco de dados, junto com as definições de tipo das classes.

Existem implementações que optam por armazenar o código em arquivos fora do banco de dados, a fim de evitar ter de integrar o software de sistemas, com compiladores, com o sistema de banco de dados.

Pode-se citar várias maneiras de fazer o armazenamento de objetos em um banco de dados:

- Uma forma é dar nomes aos objetos assim como são dados à arquivos, mas o êxito aparece somente quando se trabalha com um número relativamente moderado de objetos;
- Outra opção é alocar identificadores de objetos ou ponteiros persistentes a objetos, armazenados externamente e podem ser ponteiros físicos dentro de um banco de dados;
- Uma terceira abordagem é armazenar conjuntos de objetos e permitir que programas interajam sobre os conjuntos para encontrar os objetos requeridos. Esses conjuntos podem ser modelados como objetos de um conjunto tipo. Conjuntos tipos incluem conjuntos, multiconjuntos (isto é, conjuntos com a possibilidade da haver muitas ocorrências de um valor), listas e assim por diante. Pode-se citar um tipo especial de conjunto que é a classe extensão, representando a coleção de todos os objetos pertencentes a uma determinada classe. Classes extensão permitem que classes sejam tratadas como relações, já que é possível examinar todos os objetos da mesma, da mesma forma que se verifica as tuplas de uma relação. Além disso, quando um objeto é criado ou destruído ele é incluído ou excluído da classe extensão.

#### **2.4.2 O Paradigma Relacional Orientado a Objetos**

Diferente de linguagens de programação persistentes que acrescentam persistência e outras características de banco de dados às linguagens já existentes, como a orientação a objeto, os modelos de dados relacionais-objeto estendem o modelo relacional apresentando um modelo mais rico, incluindo orientação a objeto e acrescentando estruturas a linguagens de consulta relacionais, como SQL, para tratar os tipo de dados complexos.

Segundo Silberschatz, Korth e Sudarshan (SILBERSCHATZ, KORTH, SUDARSHAN, 1999), os sistemas de tipos aninhados permitem que os atributos de tuplas tenham tipos complexos. Tais extensões tentam preservar os fundamentos

relacionais, principalmente o acesso declaratório aos dados, enquanto estende o poder da modelagem. Sistemas de banco de dados relacionais-objeto fornecem um caminho de migração conveniente para usuários de bancos de dados relacionais que desejam usar características de orientação a objeto.

#### **2.4.2.1 Relações aninhadas**

A primeira forma normal (1NF) é tida como natural em bancos de dados convencionais, onde é necessário que todos os atributos tenham domínios indivisíveis. Mas, com os requisitos e a nova forma de modelar as mais recentes aplicações percebe-se que não há compatibilidade com a 1NF, uma vez que os usuários passam a enxergar o banco de dados como um conjunto de objetos ao invés de um conjunto de arquivos. Apesar disso, os objetos podem requerer vários arquivos para serem representados.

O modelo relacional aninhado é uma extensão do modelo relacional em que domínios podem ser definidos como atômicos ou como relações. Desta forma, o valor de uma tupla sobre um atributo pode ser uma relação e relações podem ser armazenadas dentro de relações. Logo, um objeto complexo pode ser representado por uma única tupla de uma relação aninhada. Se uma tupla de uma relação aninhada for tomada por um item de dados, conta-se uma correspondência um para um entre itens de dados e objetos na visão do usuário do banco de dados (SILBERSCHATZ, KORTH, SUDARSHAN, 1999).

Através do exemplo de uma loja de automóveis, onde existem veículos, e cada qual é composta por uma lista de acessórios, sendo registrada a data de entrada e saída da loja. O que se observa é que os domínios não são atômicos. Esta relação pode ser representada como na Tabela. 2.1:

<b>VEÍCULO</b>	<b>LISTA DE ACESSÓRIOS</b>	<b>DATA ENTRADA</b> (DIA, MÊS, ANO)	<b>DATA SAÍDA</b> (DIA, MÊS, ANO)
Astra	(direção hidráulica, vidros verdes, trio elétrico, air bag)	(12, novembro, 2000)	(04, janeiro, 2001)
Golf	(computador de bordo, barras de proteção lateral, Air bag)	(09, fevereiro, 2001)	(22, março, 2001)

**Tabela 2.1: Representação da relação Veículos**

Para organizar a relação, nos moldes atuais de banco de dados, a 1NF não poderia ser representada, pois num primeiro momento, seria necessário transformar a lista de acessórios em atributos individuais para cada possível valor. E se for considerado que data também não possui um domínio atômico (já que é constituída por dia, mês e ano), a divisão da mesma em atributos distintos também se apresentaria. Como resultado, haveria relações distintas e, conseqüentemente, o uso de junções deveria ser aplicado em consultas dependendo um custo a ser considerado.

#### **2.4.2.2 Tipos Complexos e Orientação a Objetos**

Relações aninhadas são apenas um exemplo de possíveis extensões do modelo relacional básico. Sistemas com tipos complexos e orientação a objetos tornam possível a aplicação de conceitos do modelo Entidade-Relacionamento, como identidade de entidades, atributos multivalorados, generalizações e especializações sem haver complexidade na tradução da modelagem orientada a objetos para o modelo relacional.

O que se apresenta na seqüência do texto, são exemplos com aplicação de um esboço preliminar do padrão SQL-3, onde as linguagens XSQL (uma extensão da SQL) e Illustra (que é uma versão comercial do banco de dados *Postgres*) são utilizadas, onde há a permissão do uso de tipos complexos, incluindo relações aninhadas e características de orientação a objetos.

### 2.4.2.3 Tipos Estruturados e Conjuntos

Baseado no exemplo apresentado anteriormente de relações aninhadas verifica-se, no Quadro 2.1, como é feita a declaração do mesmo utilizando extensões da SQL, a XSQL:

<b>Create type</b> <i>minhaSeqüência</i> <b>char varying</b>	← Tipo <i>minhaSeqüência</i> que é uma seqüência de caracteres de tamanho variável.
<b>Create type</b> <i>minhaData</i> ( <i>dia</i> <b>integer</b> , <i>mês</i> <b>char(10)</b> , <i>ano</i> <b>integer</b> )	← Tipo <i>minhaData</i> , constituído por dia, mês e ano.
<b>Create type</b> <i>meuVeículo</i> ( <i>nome</i> <i>minhaSeqüência</i> , <i>listaAcessórios</i> <b>setof</b> ( <i>minhaSeqüência</i> ), <i>dataEntrada</i> <i>minhaData</i> , <i>dataSaída</i> <i>minhaData</i> )	← Tipo <i>meuVeículo</i> , com os componentes <i>nome</i> , do tipo <i>minhaSeqüência</i> ; <i>listaAcessórios</i> , que é um conjunto de acessórios, cada qual do tipo <i>minhaSeqüência</i> ; e <i>dataEntrada</i> e <i>dataSaída</i> , do tipo <i>minhaData</i> .
<b>Create table</b> <i>veículos</i> <b>of type</b> <i>meuVeículo</i>	← Tabela <i>veículos</i> , com tuplas do tipo <i>meuVeículo</i> .

**Quadro 2.1: Declaração da relação Veículos utilizando XSQL através da definição de tipos**

A definição da tabela veículos difere daquelas representadas em sistemas convencionais, pois ela permite o uso de atributos estruturados (como *minhaData*) e outros que são conjuntos como *listaAcessórios*). Estas particularidades viabilizam a representação direta de atributos multivalorados e compostos do diagrama E-R. Outra questão importante a ser observada, é que a definição de tipos fica armazenada no banco de dados ficando disponível para que outras declarações façam uso das mesmas, o que não acontece em linguagens de programação persistentes, cujas definições só podem ser vistas por programas que incluem um arquivo texto com este conteúdo.

As tabelas também podem ser criadas de forma direta, sem a criação do tipo intermediário veículo. Portanto, no Quadro 2.2 está uma outra maneira de criar a tabela veículos:

```

Create table veículos
( nome minhaSeqüência,
  listaAcessórios setof(minhaSeqüência),
  dataEntrada minhaData,
  dataSaída minhaData)

```

**Quadro 2.2: Declaração da relação Veículos utilizando XSQL sem a criação de tipos**

Também é possível utilizar a definição de multiconjuntos, como exemplo, *listaImpressãoVeículos multiset(integer)*, onde vários veículos com seus respectivos números de cópias podem ser armazenados, além da definição de matrizes, tal como, *matrizAcessórios minhaSeqüência[10]*, gerando a possibilidade de identificar o n-ésimo acessório, o que não é permitido com o uso de conjuntos.

#### **2.4.2.4 Herança**

A herança pode ser aplicada tanto ao nível de tipos quanto de tabelas.

- Nível de tipos: suponha o exemplo do Quadro 2.3:

```

create type pessoa
( nome minhaSeqüência,
  cpf integer)

```

**Quadro 2.3: Declaração do tipo Pessoa**

Se for necessário modelar entidades/objetos como estudantes e professores, como é possível verificar no Quadro 2.4, os atributos nome e cpf, não precisam ser repetidos (já que ambos são pessoas), todavia são herdados, fazendo parte do conjunto de atributos da definição de estudante e de professor.

<pre> Create type estudante ( graduação minhaSeqüência   departamento minhaSeqüência) under pessoa </pre>	<pre> Create type professor ( salário integer   departamento minhaSeqüência) under pessoa </pre>
---	--

**Quadro 2.4: Herança de tipos**

No caso de monitores, que atuam como professor e estudante, podendo pertencer a departamentos distintos, também é possível fazer a modelagem



através da criação de um quarto tipo, onde se exemplifica um caso de herança múltipla, da forma apresentada no Quadro 2.5, onde o tipo monitor herda os atributos do tipo estudante e professor.

```
create type monitor
under estudante, professor
```

**Quadro 2.5: Herança múltipla de tipos**

Os atributos que tiverem o mesmo nome são renomeados a fim de evitar conflitos, usando a condição *as*. Então, redefinimos a declaração anterior no Quadro 2.6:

```
create type monitor
under estudante with (departamento as estudanteDeppto),
professor with (departamento as professorDeppto)
```

**Quadro 2.6: Renomeação de atributos com mesmo nome na aplicação de herança múltipla de tipos.**

É preciso despender muita atenção na herança de tipos, pois pode aparecer um grande número de tipos diferentes de pessoas e em muitas linguagens de programação uma entidade deve ter apenas um tipo específico, aquele subtipo de todos os tipos aos quais a entidade pertence, isto é, um aluno é do tipo estudante, mas ele não pode ser do tipo estudante e do tipo professor, a não ser que outro subtipo mais específico seja criado, no caso já criamos o subtipo monitor.

- Nível de tabelas: a herança ao nível de tabelas resolve a questão da criação de tipos mais específicos quando ocorre herança múltipla, pois é perfeitamente possível que uma tupla esteja presente em duas tabelas simultaneamente. No Quadro 2.7 está um exemplo da aplicação dessa herança:

```

create table pessoas
    (nome minhaSeqüência,
     cpf integer)

create table estudantes
    (graduação minhaSeqüência,
     departamento minhaSeqüência)
under pessoas

create table professores
    (salário integer,
     departamento minhaSeqüência)
under pessoas

```

**Quadro 2.7: Herança de tabelas**

Assim, verifica-se que as tabelas estudantes e professores herdam os atributos nome e CPF da tabela pessoas, e que, somente se houver casos de monitores com atributos específicos, não há necessidade de criar uma nova tabela para abrigá-los, pois é permitida a presença de tuplas nas tabelas estudantes e professores de uma só vez.

Apesar disso, é importante que dois requisitos básicos sejam respeitados. O primeiro diz que cada tupla presente na supertabela pessoas pode corresponder a no máximo uma tupla em cada uma das subtabelas estudantes e professores. Isso garante que não haja repetição de informações da mesma pessoa, fato não condizente com a realidade. O segundo requisito exige que cada tupla existente nas subtabelas deve corresponder a exatamente uma tupla na supertabela, ou seja, sempre haverá um estudante ou professor que está ligado a uma única tupla na tabela pessoas, algo óbvio no mundo real, mas que deve ser garantido no banco de dados, para que não um estudante ou professor encontre várias ocorrências ou a ausência, o que resultaria um erro.

A herança múltipla também é possível, pois como já dissemos uma tupla pode pertencer a duas tabela ao mesmo tempo, mas pode ser explicitada como no Quadro 2.8:

```

create table monitores
    under estudantes with departamento as estudantesDeppto,
    under professores with departamento as professoresDeppto

```

**Quadro 2.8: Herança múltipla de tabelas**

As exigências aqui também se fazem presentes. É imprescindível que a tupla presente em monitores esteja na tabela estudantes e na tabela professores de forma simultânea.

#### **2.4.2.5 Tipos Referência**

Linguagens orientadas a objetos tornam possível a referência a objetos. Um atributo de um determinado tipo pode ser uma referência a um objeto de um tipo especificado. Pode-se usar o exemplo de livros, cujos autores são representados por um atributo listaAutores. Este atributo pode ser definido como:

- `listaAutores setof(minhaSeqüência)`, utilizando o tipo `minhaSeqüência` definido em exemplos anteriores; ou,
- `listaAutores setof(ref(pessoa))`, usando o tipo referência, onde o atributo é um conjunto de referências a objetos `pessoa`.

#### **2.4.2.6 Consultas com Tipos Complexos**

Nesta sessão serão apresentadas formas de consultar o banco de dados usando a XSQL. Começando com uma consulta simples, fazendo uso do exemplo sobre veículos: encontrar o nome e o ano de entrada de cada veículo. A consulta fica como está no Quadro 2.9, onde se observa a notação de ponto para acessar o campo *ano* do atributo *dataEntrada*.

```
select nome, dataEntrada.ano
from veículos
```

**Quadro 2.9: Representação de tipos complexos**

##### ***2.4.2.6.1 Atributos Relação-Valorados***

Uma expressão para relação é permitida em qualquer parte da consulta em que o nome da relação pode aparecer, como na condição *from*. Suponha as declarações do Quadro 2.10:

<pre> create type acessório (nome minhaSeqüência fabricante minhaSeqüência) </pre>	<pre> create table veículosObj (nome minhaSeqüência listaAcessórios setof(ref(acessório)), dataEntrada minhaData, dataSaída minhaData) </pre>
--	---

**Quadro 2.10: Tipo Acessório e Tabela VeículosObj**

Se uma listagem contendo todos os veículos com o acessório “air bag” precisar ser emitida, a consulta deverá ser como a do Quadro 2.11:

<pre> select nome from veículosObj where “air bag” in listaAcessórios </pre>
--

**Quadro 2.11: Consulta sobre atributo relação-valorado**

Note-se que na SQL convencional, a expressão onde o atributo relação-valorado listaAcessórios aparece, seria exigido o uso de uma subexpressão *select-from-where*, o que não é necessário com o uso da SQL estendida.

No caso de ser necessário um relatório contendo pares com o nome de cada veículo e acessórios de forma individual, a consulta ficaria como no Quadro 2.12 abaixo:

<pre> select V.nome, A.nome from veículosObj as V, V.listaAcessórios as A </pre>
--

**Quadro 2.12: Consulta sobre atributo relação-valorado na SQL estendida**

Funções agregadas, como min, max, count, average, etc, que consultam vários valores e retornam apenas um único valor como resultado, também podem ser aplicadas a qualquer expressão relação-valorada. Então, se for requerido um relatório com o nome e o número de acessórios para cada veículo (Quadro 2.13), a consulta poderia ser escrita como segue:

<pre> select nome, count(listaAcessórios) from veículosObj </pre>
---

**Quadro 2.13: Consulta sobre atributo relação-valorado na SQL estendida**

### 2.4.2.6.2 Expressões Path

A notação de ponto, usada para referir-se a atributos compostos, também pode ser usada com referências. No exemplo a seguir, duas tabelas são utilizadas: a tabela *peessoas*, definida anteriormente, e a tabela *mestrandos* definida abaixo. A consulta desejada é: encontrar o nome dos orientadores de todos os mestrandos e está descrita no Quadro 2.14.

<pre>create table mestrandos (orientador ref(peessoas)) under pessoas</pre>	<pre>select mestrandos.orientador.nome from mestrandos</pre>
---	--

**Quadro 2.14: Expressões path**

Como “*mestrandos.orientador*” é uma referência à tupla na tabela *peessoa*, o atributo *nome* apresentado na consulta é o atributo *nome* da tupla referenciada da tabela *peessoa*.

A vantagem de usar referências desse tipo está na simplificação da consulta, dispensando o uso de junções entre tabelas através de chaves estrangeiras.

A expressão da forma “*mestrandos.orientador.nome*” é chamada expressão *path*. Cada atributo na expressão *path* tem um único valor, ou seja, uma referência no caso do exemplo adotado acima.

### 2.4.2.7 Criação de Objetos Complexos e Valores

Um objeto é criado a partir de uma função *constructor* baseada em seu tipo, ou seja, um objeto do tipo Veículo é criado através de sua função *constructor* Veículo( ). Quando ela é chamada, um novo objeto é criado, o campo *oid* (identificador do objeto) é preenchido e o objeto é retornado. Depois, os campos do objeto devem ser preenchidos explicitamente.

Os valores são anexados ao objeto usando a função *insert into*, como mostrado no Quadro 2.15. Desta maneira, ao objeto veículos podem ser relacionados os seguintes valores:

**insert into veículos**

**values**(“Astra”, set(“direção hidráulica”, “vidros verdes”, “trio elétrico”, “air bag”), (12, “novembro”, 2000), (04, “janeiro”, 2001))

**insert into veículos**

**values**(“Golf”, set(“computador de bordo”, “barras de proteção lateral”, “air bag”), (09, “fevereiro”, 2001), (22, “março”, 2001))

**Quadro 2.15: Atribuição de valores aos objetos através da instrução *Insert Into***

Resultando, esquematicamente na Tabela 2.3 (repetição da tabela 2.1 – página 43 – por questões de praticidade de leitura):

<b>VEÍCULO</b>	<b>LISTA DE ACESSÓRIOS</b>	<b>DATA ENTRADA</b> (DIA, MÊS, ANO)	<b>DATA SAÍDA</b> (DIA, MÊS, ANO)
Astra	(direção hidráulica, vidros verdes, trio elétrico, air bag)	(12, novembro, 2000)	(04, janeiro, 2001)
Golf	(computador de bordo, barras de proteção lateral, Air bag)	(09, fevereiro, 2001)	(22, março, 2001)

**Tabela 2.3: Representação da relação Veículos apresentando os dados incluídos com a operação *Insert Into***

A operação de atualização em objetos complexos é feita de forma similar à atualização em relações convencionais, que estão na 1NF.

Sistemas de banco de dados dos tipos estudados neste trabalho estão presentes no mercado. Fica a critério do projetista de banco de dados escolher aquele que mais se adapta ao estilo do seu projeto.

O fato da linguagem SQL ser declaratória e limitada em alguns aspectos em comparação a outras linguagens de programação, faz com que os dados estejam relativamente protegidos no que diz respeito a erros de programação, além de facilitar a realização de otimizações de alto nível, como redução de I/O. Apesar disso, em aplicações que realizam um grande número de acessos ao banco de dados e executam na

memória principal, o uso de tal linguagem pode comprometer seriamente o seu desempenho.

Mas, para estes tipos de aplicações, com alta frequência de acesso ao banco de dados, as linguagens de programação persistentes se apresentam como fortíssimas candidatas, pois fornecem acesso com baixo overhead ao dado persistente e eliminam a tradução de tipos de dados se for necessário o uso de uma outra linguagem para a manipulação dos mesmos. Entretanto, os possíveis erros de programação podem trazer grandes possibilidades de corrupção dos dados, além de não apresentarem muita flexibilidade no momento de realizar consultas (SILBERSCHATZ, KORTH, SUDARSHAN, 1999).

Portanto, pode-se concluir que sistemas relacionais são mais apropriados para tipos de dados simples, havendo um grande poder nas linguagens de consulta e alta proteção dos dados. As linguagens de programação simples baseadas em BDOOs, são melhor utilizadas quando tipos de dados complexos estão envolvidos, fornecendo uma boa integração com linguagens de programação, sem falar do alto desempenho. Já sistemas relacionais-objeto, trabalham com tipos de dados complexos, oferecendo uma grande flexibilidade na realização de consultas aliada à alta proteção dos dados.

## **2.5 Modelagem Lógica dos Dados**

Os *softwares* têm participado cada vez mais do cotidiano das pessoas, seja na área pessoal ou profissional. E existem muitas aplicações que se responsabilizam não só por processos burocráticos, mas pelo bem estar da vida humana. Em função disso, é cada vez mais importante que os sistemas sejam cuidadosamente desenvolvidos, desde a coleta dos dados do mundo real até a geração do produto final (BOOCH, 2002).

Segundo *Larman* (LARMAN, 2000), para criar uma aplicação é necessário ter tanto descrições abrangentes quanto descrições mais detalhadas do problema, incluindo a maneira que requisitos e restrições são atendidos. Para que isso ocorra, é preciso fazer uma investigação minuciosa, além de apresentar uma solução lógica à questão discutida, ou seja, dizer como o sistema deve atender os requisitos.

A qualidade de um *software* não acontece simplesmente, é preciso usar de métodos organizacionais para que ela se faça presente. Ultimamente, se tem usado métodos de terceira geração para especificar, visualizar e documentar os dados colhidos do mundo real, segundo o paradigma da orientação a objetos (BOOCH, 2002). É aí que entra a UML – Linguagem Unificada de Modelagem – que é uma linguagem utilizada para modelar logicamente os dados de um sistema com base em conceitos da programação orientada a objetos. Ela unifica principalmente os métodos de *Booch*, *Rumbaugh* e *Jacobson*, mas tem uma amplitude maior e hoje constitui um padrão OMG (*Object Management Group*) (FOWLER, SCOTT, 2000). A razão fundamental para usar UML está estreitamente relacionada com a facilidade de comunicação e entendimento entre profissionais envolvidos num projeto, além de simplificar o debate e compreensão entre desenvolvedores e usuários da aplicação.

O que mais chama atenção nessa linguagem, são as diversas técnicas que podem ser aplicadas de acordo com o objetivo da equipe de desenvolvimento num dado momento do projeto. São técnicas que representam dados de maneira formal, padronizada, expressiva e flexível (LARMAN, 2000). Não existe um processo de desenvolvimento a ser seguido, pois diversos fatores estão envolvidos, como o tipo da aplicação e a quantidade de profissionais da equipe de desenvolvimento, entre outros (FOWLER, SCOTT, 2000). As ferramentas dessa linguagem são empregadas conforme a necessidade encontrada em situações distintas da análise e projeto de um sistema, não havendo obrigação de aplicar todas as técnicas. Portanto, na seqüência estão descritos os artefatos mais comumente utilizados.

### **2.5.1 Definição dos Requisitos**

Numa visão macro, UML coloca o conceito de modelo do sistema, através do qual são organizados e descritos detalhes do mundo real no qual a aplicação abordada está inserida, incluindo elementos coesos e fortemente relacionados. Modelos são constituídos por diagramas e documentos que descrevem coisas. Eles são construídos independente de qualquer consideração física posterior (Conceptual, Logical..., 2002). De acordo com *Larman* (LARMAN, 2000), os mesmos podem ser classificados em:



*estáticos*, que descrevem propriedades estruturais, como o diagrama de classes; e *dinâmicos*, que esclarecem propriedades comportamentais de um sistema, como por exemplo, o diagrama de interação.

O modelo de sistema, por sua vez, é dividido em modelo de análise e modelo de projeto. No primeiro, é feita uma investigação do escopo do problema-alvo, onde é feita a **definição dos requisitos**, ou seja, dos objetivos gerais, dos usuários/clientes, das funções e dos atributos do sistema (Quadro 2.16). Para facilitar o entendimento, um sistema exemplo muito simples denominado “Sistema de Controle de Veículos” será utilizado para demonstrar a aplicação dos artefatos UML nas fases de análise e projeto de um sistema.

<p><b>Objetivos gerais</b> O sistema deve armazenar e fornecer as informações relativas a um veículo, registrando sua entrada e saída da empresa.</p> <p><b>Clientes/Usuários</b> Empresas de revenda de veículos.</p> <p><b>Funções</b></p> <ul style="list-style-type: none"> <li>✓ Cadastrar os dados de um veículo</li> <li>✓ Registrar os dados da entrada do veículo na empresa</li> <li>✓ Registrar os dados da saída do veículo da empresa</li> <li>✓ Listar os veículos em estoque e os que foram vendidos, na totalidade ou num período.</li> <li>✓ Emitir documentos relativos à transação de compra e venda de veículos e de controle interno da empresa.</li> </ul>	<p><b>Atributos</b></p> <ul style="list-style-type: none"> <li>✓ Marca</li> <li>✓ Placa</li> <li>✓ Ano</li> <li>✓ Km</li> <li>✓ Código RENAVAN</li> <li>✓ Número CHASSI</li> <li>✓ Cor</li> <li>✓ Combustível</li> <li>✓ Nome do proprietário na entrada</li> <li>✓ CPF/CNPJ do proprietário na entrada</li> <li>✓ Funcionário responsável pelo recebimento</li> <li>✓ Data e hora de entrada</li> <li>✓ Valor pago, gastos, custo final e valor de venda.</li> <li>✓ Nome do proprietário na saída</li> <li>✓ CPF/CNPJ do proprietário do veículo na saída</li> <li>✓ Funcionário responsável pela entrega</li> <li>✓ Data e hora de saída</li> </ul>
--	--

**Quadro 2.16: Requisitos básicos de um sistema**

## 2.5.2 Casos de Uso – Uses case

Também faz parte da análise a elaboração de **Casos de Uso** (Uses Case – UC), que são documentos escritos em prosa contendo histórias ou descrições de cenários na utilização do sistema (LARMAN, 2000). Segundo *Fowler e Scott* (FOWLER, SCOTT, 2000), um cenário é “uma seqüência de passos que descreve uma interação entre um

usuário e um sistema”. Caso de uso, por sua vez, também pode ser entendido como um conjunto de cenários vinculados por um objetivo comum do ponto de vista do usuário.

Não existe um modelo único para a organização de um caso de uso. A equipe de trabalho pode simplesmente enumerar frases que descrevem as etapas de uma atividade (FOWLER, SCOTT, 2000) ou utilizar-se de itens que sejam relevantes na descrição de cada sistema abordado a fim de auxiliar na montagem do texto, como exemplo, atores, finalidade, descrição ou fluxo principal, pré-condições, pós-condições, fluxos alternativos, documentos relacionados, entre outros (LARMAN, 2000). Para facilitar a identificação dos atores participantes de um caso de uso, o projetista tem mais êxito quando pensa nos papéis desempenhados no sistema, ao invés de centralizar-se em pessoas ou em cargos.

Nos Quadros 2.17.a e 2.17.b, são encontrados os itens escolhidos para a elaboração dos casos de uso do sistema exemplo, onde o cenário principal está descrito no item *Fluxo normal da tarefa* e cenários alternativos são relatados no item *Desvio de fluxo*. Os casos de uso destacados para o sistema exemplo são: cadastro de veículos (Quadro 2.17.a), entrada de veículos no estoque (Quadro 2.17.b), saída de veículos da empresa, cadastro de funcionários, cadastro de clientes, alteração de dados de veículos, exclusão de veículos, localização de veículos, localização de funcionários, localização de clientes, impressão de relatórios. Escolheu-se demonstrar apenas dois casos de uso por questões de simplificação.

<b>Caso de Uso 1 (Use Case 1 – UC-01)</b>	
<u>Referência</u>	<b>UC-01</b>
<u>Nome</u>	Cadastro de veículos.
<u>Atores</u>	Secretária e vendedores
<u>Breve descrição</u>	Quando um veículo é adquirido pela empresa, seus dados (como marca, placa, ano, cor, combustível, etc.) são registrados no sistema.
<u>Pré-condições</u>	Conhecimento do número da placa do veículo.
<u>Fluxo normal da tarefa</u>	O funcionário solicita um novo registro, digita os dados do veículo e cadastra todas as informações.
<u>Desvio de fluxo</u>	Quando o ator digita a placa do veículo e esta já estiver registrada para outra unidade, o sistema oferece a possibilidade de visualizar o item referente àquele número de placa. Mas, se o usuário perceber que foi apenas um erro de digitação, opta por seguir informando os demais dados do novo veículo.
<u>Documento(s) relacionado(s)</u>	Inexistente até o momento.

**Quadro 2.17.a: Caso de uso Cadastro de Veículos.**

<b>Caso de Uso 2 (Use Case 2 – UC-02)</b>	
<u>Referência</u>	UC-02
<u>Nome</u>	Entrada de veículos no estoque.
<u>Atores</u>	Secretária e Proprietário da empresa.
<u>Breve descrição</u>	O usuário digita as informações referentes ao proprietário do veículo (fornecedor) que está sendo adquirido pela empresa, data, horário, funcionário e também dados que dizem respeito aos custos envolvidos nessa aquisição.
<u>Pré-condições</u>	O veículo deve estar cadastrado no sistema.
<u>Fluxo normal da tarefa</u>	O usuário digita ou localiza o proprietário do veículo (fornecedor), informa a data e hora da entrada, escolhe o funcionário responsável pelo recebimento do veículo, digita o valor pago pelo mesmo, o gasto, o custo final e o valor de venda, e finaliza gravando a inclusão dessa unidade no estoque.
<u>Desvio de fluxo</u>	<ol style="list-style-type: none"> <li>1. Se o proprietário não estiver registrado no sistema, o usuário deve registrá-lo, no lugar apropriado e, em seguida inclui-lo na ficha de entrada no estoque.</li> <li>2. Se o funcionário não estiver registrado no sistema, o usuário deve cadastrá-lo, no lugar apropriado e, em seguida inclui-lo na ficha de entrada no estoque.</li> </ol>
<u>Documento(s) relacionado(s)</u>	UC -01

**Quadro 2.17.b: Caso de uso Entrada de veículos no estoque**

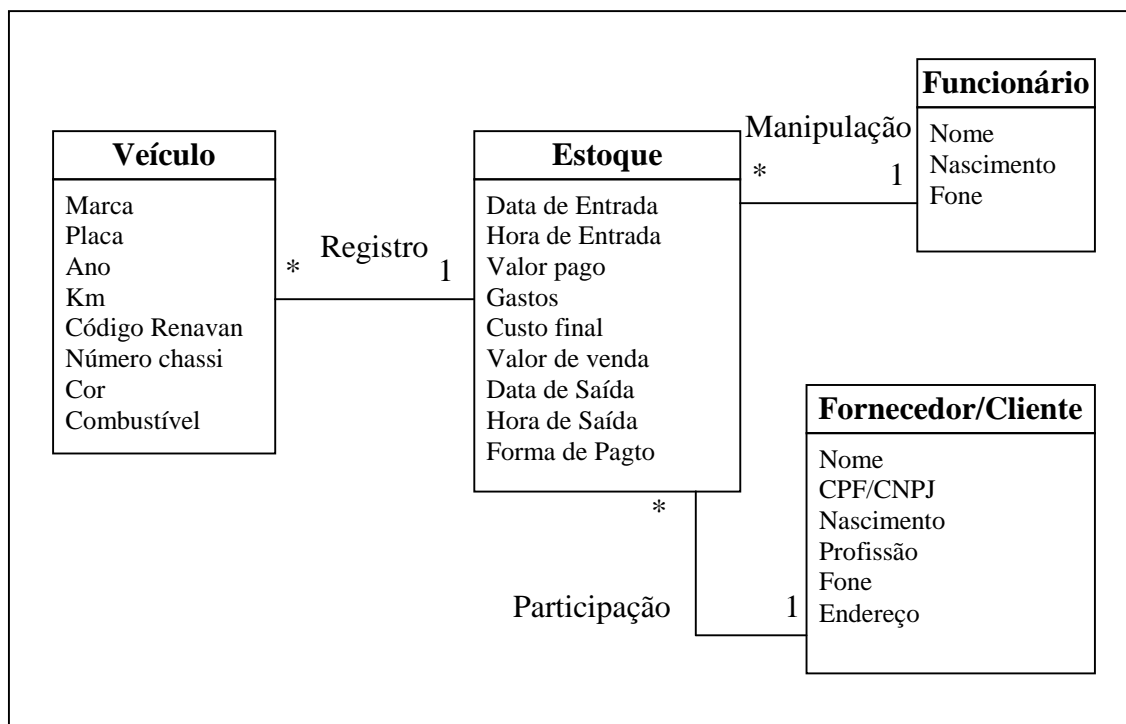
### 2.5.3 Modelo Conceitual

Outra ferramenta utilizada na fase de análise é a elaboração de um **modelo conceitual**, que de acordo com *Larman* (LARMAN, 2000), é uma representação de conceitos em um domínio do problema. Ele também coloca que um conceito é uma idéia, uma coisa ou um objeto, isto é, o mapeamento de uma informação estática do mundo real.

Aliás, é nesta fase que está a diferença fundamental entre a análise estruturada e a análise orientada a objetos. Na primeira, é feita uma divisão do problema por funções e na segunda a divisão é feita por conceitos, que mais tarde ajudarão na distinção das classes. Mas, é importante que se diga que os conceitos listados nesta fase são independentes de implementação, ou seja, não há uma obrigatoriedade de que os mesmos tornem-se classes mais na frente.

Uma forma de construir o modelo conceitual é utilizar-se dos passos seguintes (LARMAN, 2000), sendo que o resultado final pode ser visto na Figura 2.4:

1. Listar os conceitos candidatos identificando substantivos nos casos de uso descritos ou utilizando-se da *Lista de Categoria de Conceitos* (ver Anexo1);
2. Desenhar os conceitos destacados no passo 1 em um diagrama conceitual, representados graficamente, de forma individual, através de um retângulo;
3. Acrescentar as associações necessárias de forma a registrar os relacionamentos existentes entre os conceitos destacados, onde a representação é feita por uma linha entre os conceitos, com um nome e a indicação da quantidade de conceitos participantes da associação (multiplicidade); e,
4. Acrescentar os atributos a fim de completar os requisitos de informação.



**Figura 2.8: Modelo Conceitual do "Sistema Controle de Veículos"**

## 2.5.4 A Ferramenta Glossário

É importante que seja usada a ferramenta **Glossário** em paralelo à realização das outras técnicas da linguagem de modelagem. Trata-se de um documento simples, onde os termos do sistema que precisam de esclarecimentos são descritos, pois dessa forma facilita o entendimento entre os membros da equipe de desenvolvimento e reduz as possibilidades de mal-entendidos.

## 2.5.5 Diagrama de Seqüência do Sistema

Há mais uma técnica útil na fase de análise, é o **Diagrama de Seqüência do Sistema**. Ele trabalha com informações dinâmicas do problema-alvo e, portanto, objetiva ilustrar os eventos gerados pelos atores que são reconhecidos pelo sistema (LARMAN, 2000). É elaborado com base nos casos de uso relatados anteriormente. Um exemplo pode ser visto na Figura 2.5, onde se encontra o diagrama de seqüência do sistema para o caso de uso Cadastro de veículos.

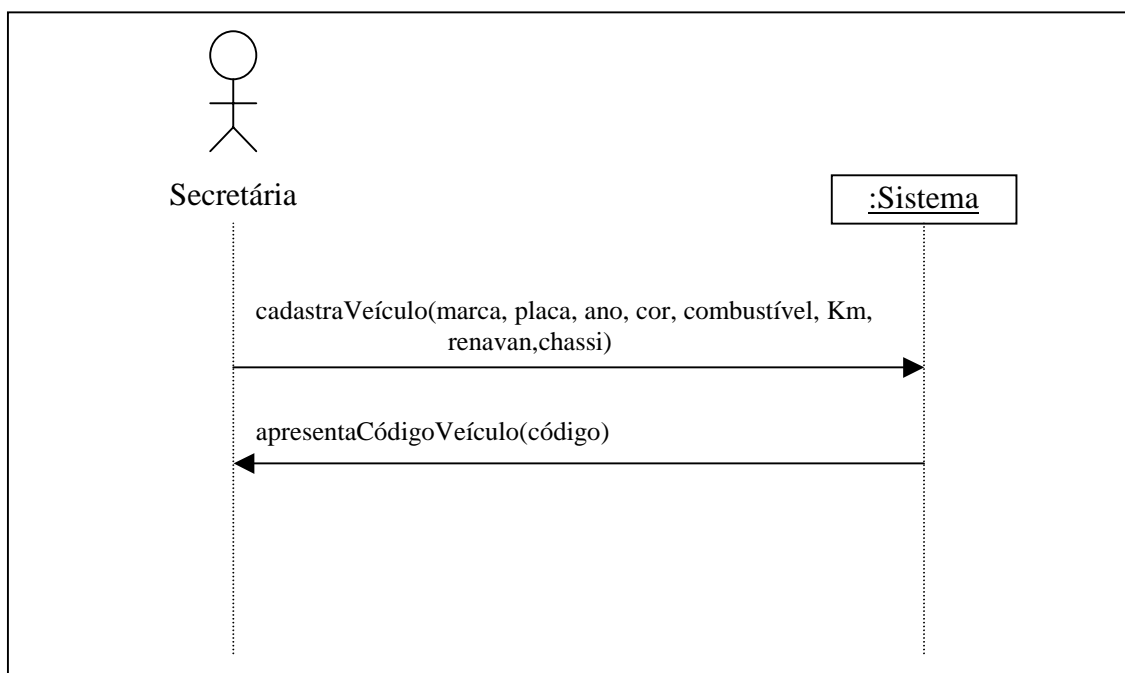
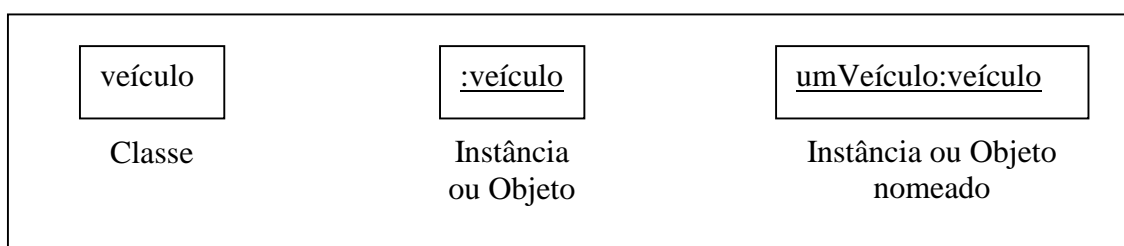


Figura 2.9: Diagrama de Seqüência do Sistema para o Caso de Uso Cadastro de Veículos

## 2.5.6 Diagrama de Interação

Durante a análise, dá-se mais atenção à compreensão e definição dos requisitos, dos conceitos e das operações. Já na fase de projeto, a preocupação está voltada para as soluções lógicas que são apresentadas ao problema analisado. Nessa etapa, os esforços estão sobre a dinâmica das informações, onde o artefato principal que ilustra a maneira com que os objetos comunicam-se e atendem aos requisitos é o **diagrama de interação** (LARMAN, 2000).

Antes de dar seguimento, é preciso que se faça um esclarecimento sobre a nomenclatura e simbologia referentes à classe, instância ou objeto e instância ou objeto nomeado, que será utilizada deste ponto em diante nos diagramas que compõem a fase de projeto do sistema. A Figura 2.6 demonstra, segundo *Larman* (LARMAN, 2000), que os três itens são representados por um retângulo, a diferença está na forma com que o nome é colocado dentro desta forma geométrica: a classe tem apenas seu nome escrito; a instância ou objeto tem o nome da classe sublinhado e antecedido pelo sinal de “:” (dois pontos); e a instância ou objeto nomeado tem o nome dado à instância seguido pelo sinal de “:” (dois pontos) e pelo nome da classe, sendo que tudo é sublinhado.

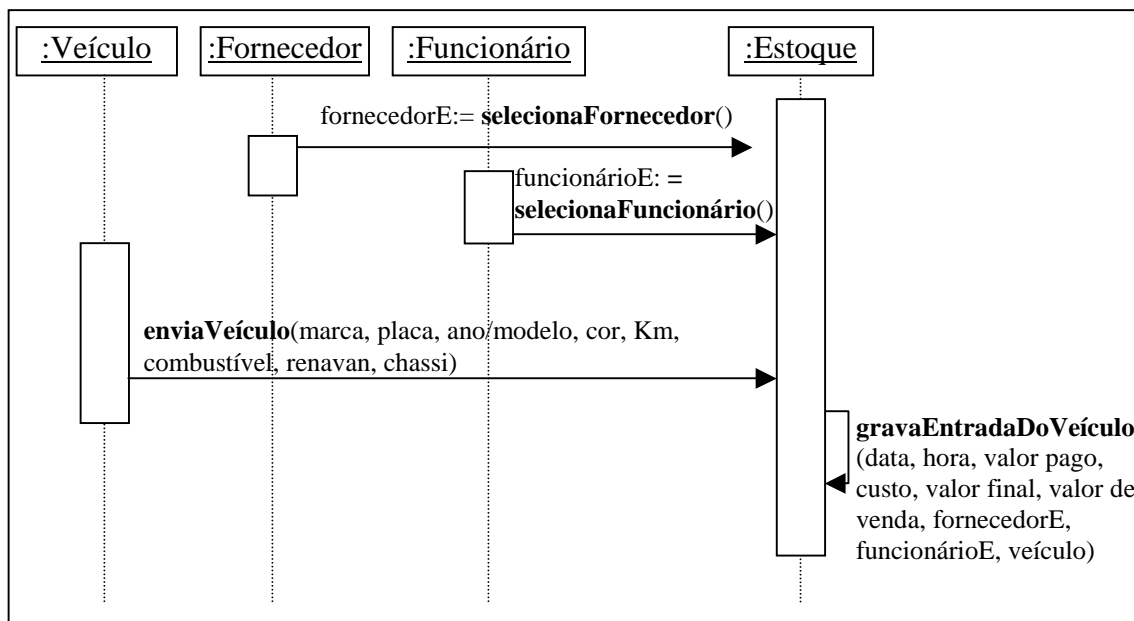


**Figura 2.10: Nomenclatura e simbologia parcial de classes e instâncias ou objetos**

Existem dois tipos de diagrama de interação: o **diagrama de seqüência** e o **diagrama de colaboração**. Não há necessidade da aplicação dos dois tipos, pois ambos têm a mesma finalidade: demonstrar o fluxo de mensagens trocadas entre objetos durante a execução de uma determinada operação. O importante é verificar qual deles melhor se aplica ao sistema estudado, embora a utilização de ambos pode ajudar quando

se tratar de sistemas complexos. Cada diagrama de interação diz respeito a um único caso de uso (FOWLER, SCOTT, 2000).

No diagrama de seqüência, como é possível conferir na Figura 2.7, os objetos são colocados lado a lado em retângulos e abaixo de cada um é colocada uma linha pontilhada vertical que indica o seu tempo de vida útil. Nestas linhas, pode-se utilizar caixas de ativação, para indicar quando o objeto está ativo. As mensagens trocadas entre os objetos são representadas por uma flecha horizontal entre as linhas de vida, devidamente nomeadas, podendo ou não conter parâmetros. A leitura das mensagens no diagrama é feita de cima para baixo (FOWLER, SCOTT, 2000). A vantagem do diagrama de seqüência está na possibilidade de observar o fluxo de informações de uma forma global. É possível demonstrar nesse tipo de diagrama diversas situações da execução de uma operação, como autochamada (quando um objeto envia uma mensagem para si mesmo), condição, retorno, processos concorrentes, remoção de objeto, iteração, entre outros.



**Figura 2.11: Diagrama de Seqüência para o Caso de Uso Entrada de Veículos no Estoque**

O diagrama de colaboração dispõe as classes como ícones e as mensagens também são representadas por flechas. A diferença está na leitura que é feita seguindo a numeração das mensagens. Essa maneira dificulta a leitura global das mensagens, mas consegue demonstrar o relacionamento entre os objetos (FOWLER, SCOTT, 2000). Este tipo de diagrama de interação tem grande capacidade de demonstrar a troca de

mensagens, expressar claramente informações contextuais, além de economizar espaço (LARMAN, 2000). A Figura 2.8 demonstra um exemplo para o caso de uso Entrada de Veículos no Estoque.

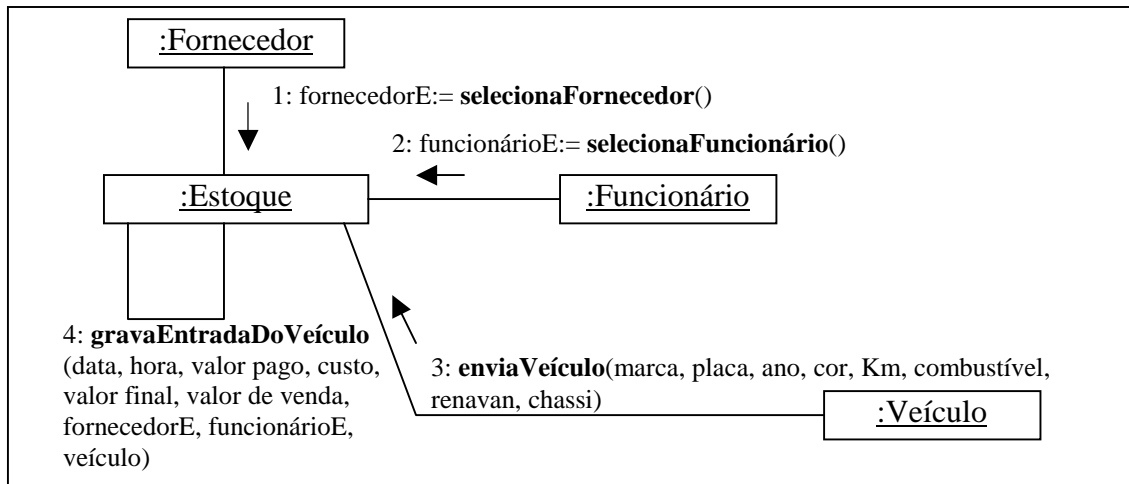


Figura 2.12: Diagrama de Colaboração para o Caso de Uso Entrada de Veículos no Estoque

### 2.5.7 Diagrama de Classes

Agora que os diagramas de interação já foram elaborados, é possível passar para a construção do **diagrama de classes**, onde é colocada a definição das entidades de *software*, ao invés dos conceitos do mundo real que fazem parte do modelo conceitual. Neste diagrama, as classes são desenhadas da forma que aparece na Figura 2.9, é uma caixa dividida em três sessões: a porção superior contém o nome da classe, na do meio são encontrados os atributos da mesma e, na parte inferior estão relacionados os métodos da classe (LARMAN, 2000).

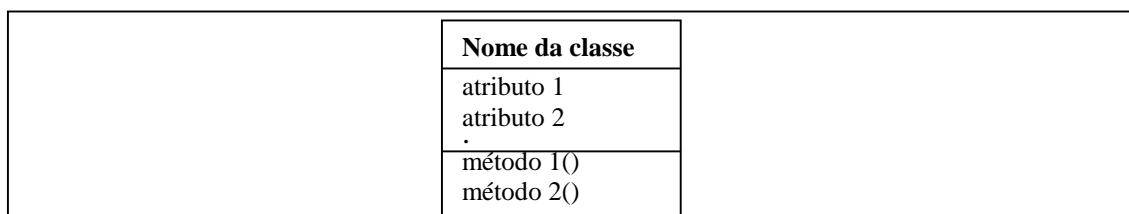


Figura 2.13: Nomenclatura e simbologia UML de uma Classe

Um diagrama de classes, segundo *Fowler e Scott* (FOWLER, SCOTT, 2000), expõe os tipos de objetos do sistema e as relações estáticas entre eles, que são as associações. As associações têm duas pontas, cada qual ligada às classes inter-

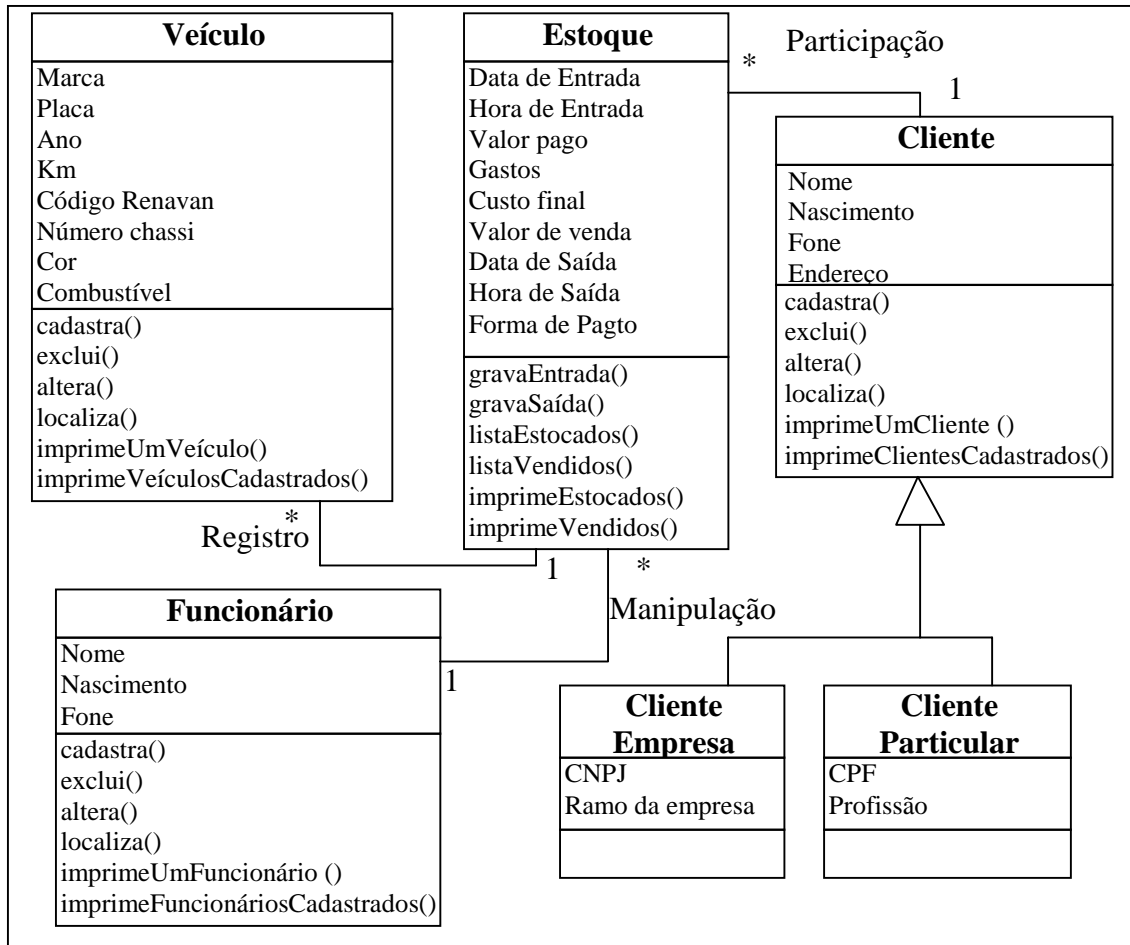


relacionadas, onde é colocada a quantidade de objetos que podem fazer parte do relacionamento indicado; isto é conhecido como multiplicidade. As indicações mais comumente encontradas são:

- “\*” mostrando que **0 ou muitos objetos** podem participar da associação;
- “1” dizendo que **somente 1 objeto** pode fazer parte do relacionamento; e,
- “0..1” representando o envolvimento de **0 ou 1 objeto**.

Dependendo do sistema, multiplicidades mais específicas, onde se sabe o número exato mínimo e máximo de participação dos objetos, também podem ser aplicadas. Normalmente, as associações recebem nomes que facilitam a leitura, mas isso não é obrigatório. Os atributos indicam as características de uma classe e armazenam valores correspondentes. Dependendo do nível de detalhe do diagrama, pode ser incluído além do nome, o tipo, a multiplicidade e valor-por-omissão de cada atributo, no seguinte formato: *nome [multiplicidade]: tipo = valor-por-omissão* (FOWLER, SCOTT, 2000). Quanto aos métodos, pode-se dizer que são processos que uma classe sabe realizar. É a forma que os objetos têm de se comunicarem entre si. Através deles, os atributos são consultados ou modificados por si ou por outros objetos que tenham permissão (LEMAY, 1998).

Sendo assim, na Figura 2.10 é encontrado o diagrama de classes para o sistema exemplo adotado – “Sistema de Controle de Veículos”.



**Figura 2.14: Diagrama de Classes do “Sistema Controle de Veículos”**

No diagrama de classes, também é possível modelar relacionamentos de generalização e especialização, agregação e composição, classificação múltipla e dinâmica, interfaces e classes abstratas, objetos de referência e objetos de valor, *frozen*, associações qualificadas, classes de associação, classe parametrizada, visibilidade, entre outros. São conceitos ditos avançados por *Fowler e Scott* (FOWLER, SCOTT, 2000), que são aplicados em sistemas bem mais complexos que o exemplo utilizado como referência para a demonstração dos artefatos UML e, portanto, não serão discursados aqui. Para uma visão mais detalhada sobre esses conceitos, pode-se consultar a obra de *Fowler e Scott* (FOWLER, SCOTT, 2000).

## 2.5.8 Pacotes

Esta técnica é usada quando se trabalha em um sistema grande, onde a organização do material que compõe a análise e projeto torna-se dificultosa em função do espaço requerido. Portanto, os pacotes são um artifício usado para agrupar os componentes UML, inclusive outros pacotes, mas os diagramas de classes são reunidos com mais frequência. Desta forma, é possível obter um nível mais global de entendimento do sistema abordado (FOWLER, SCOTT, 2000). A notação e simbologia de um pacote podem ser observadas na Figura 2.11. Verifica-se que o nome é colocado no “rótulo”, quando o conteúdo do pacote é apresentado ou no interior do mesmo, caso contrário.

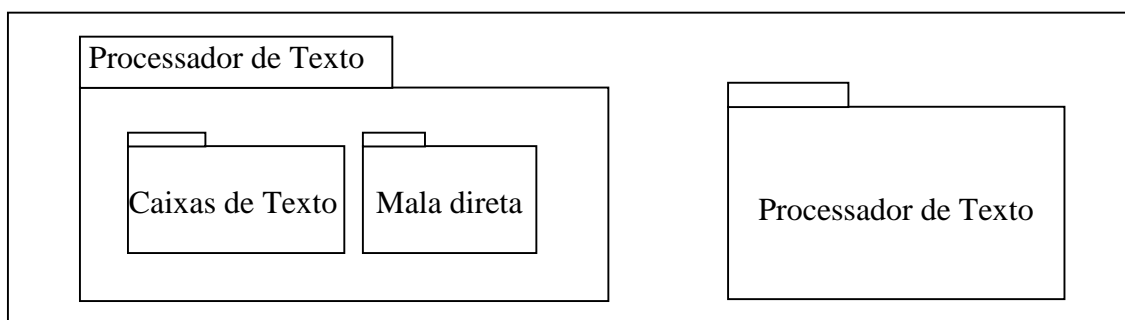


Figura 2.15: Nomenclatura e simbologia de Pacotes

## 2.5.9 Diagrama de Estados

É uma técnica que ilustra o comportamento de um sistema, enfocando cada objeto em todos os estados possíveis que pode assumir durante o funcionamento de uma determinada aplicação, em função dos eventos ocorridos sobre ele ou a partir dele (FOWLER, SCOTT, 2000). Com este artefato é possível verificar como um único objeto se comporta em vários casos de uso.

Na Figura 2.12, observa-se o diagrama de estados para um veículo do sistema exemplo adotado. Nele são colocados os estados possíveis que um veículo pode assumir e as transições que o levam de um estado a outro. O estado inicial é indicado por um círculo preto preenchido. Os demais são representados por retângulos com cantos

arredondados, devidamente nomeados. Opcionalmente pode haver a indicação de atividade realizada em um determinado estado, escrita no formato: *do/atividade*. A sintaxe para as transições é formada por três porções, todas opcionais: *evento [guarda] / ação*. Um *guarda* é uma condição lógica que retornará somente “verdadeiro” ou “falso” e, portanto uma transição guardada só ocorre se o guarda for “verdadeiro” (FOWLER, SCOTT, 2000).

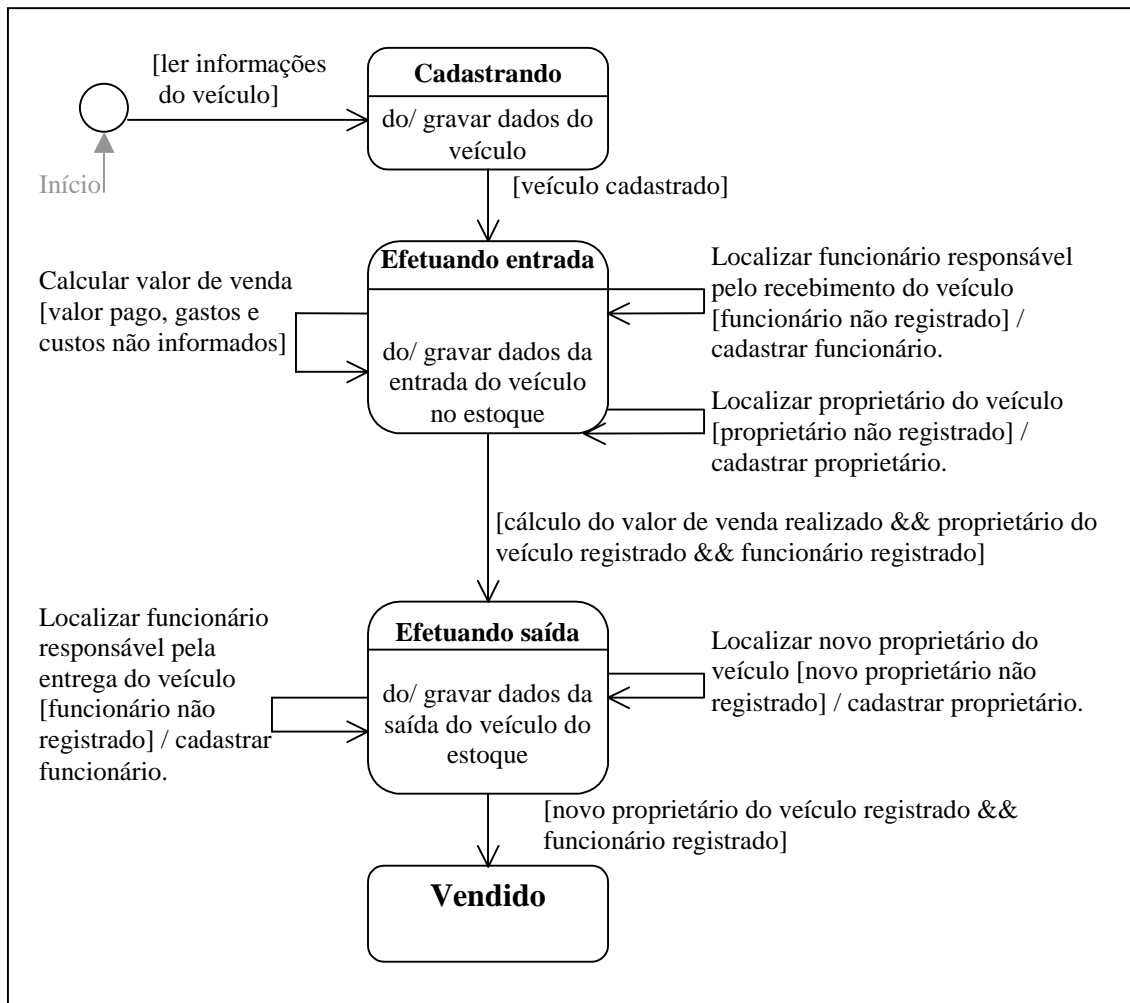


Figura 2.16: Diagrama de Estados para um Veículo

### 2.5.10 Diagrama de Atividades

O diagrama de atividades é essencialmente útil para demonstrar atividades que podem ser feitas em paralelo, sem preocupar-se com a seqüência de execução. A modelagem de processos concorrentes também é beneficiada por esta técnica, pois é possível projetar graficamente os caminhos conhecidos e o momento em que precisam

ser sincronizados (FOWLER, SCOTT, 2000). Neste diagrama também se pode descrever comportamentos condicionais da seqüência de uma atividade. A desvantagem desse artefato é que não há possibilidade de modelar colaborações entre objetos. Por isso, o diagrama de atividades tem mais utilidade quando aplicado junto a outras técnicas da linguagem. Trata-se de uma variante do diagrama de estados, logo a nomenclatura adotada é bastante semelhante. A diferença principal está na representação das atividades que é feita através de um retângulo com laterais arredondadas, ao invés de ser somente nos cantos.

*Fowler e Scott* (FOWLER, SCOTT, 2000) diz que, comportamentos condicionais são indicados por **desvios** (*branch*) e **intercalações** (*merge*). Os primeiros são losangos com uma única entrada e várias transições de saída mutuamente exclusivas, pois somente uma poderá ser seguida. Já as intercalações, também indicadas por losangos, têm várias entradas e apenas uma saída para indicar o final da condição marcada por um desvio.

Outros elementos que compõe o diagrama de atividades são as **separações** (*fork*) e as **junções** (*joins*), ambas representadas por uma pequena barra horizontal. Uma separação recebe apenas uma transição de entrada e dela saem várias transições que são realizadas em paralelo ou de forma intercalada, não importando a seqüência. As junções têm a função de indicar que a transição seguinte somente será efetuada se as transições de entrada tiverem completado seus objetivos. Uma junção é usada para finalizar uma separação, modelando assim atividades concorrentes que precisam ser sincronizadas.

Também é possível modelar atividades mais complexas fazendo uso de outros componentes desse diagrama, como decomposição de atividades, concorrência dinâmica e raias (*swimlanes*), que não serão aplicados aqui devido à simplicidade do sistema exemplo adotado.

Na Figura 2.13, está o diagrama de atividades para veículos, onde os componentes citados acima podem ser observados.

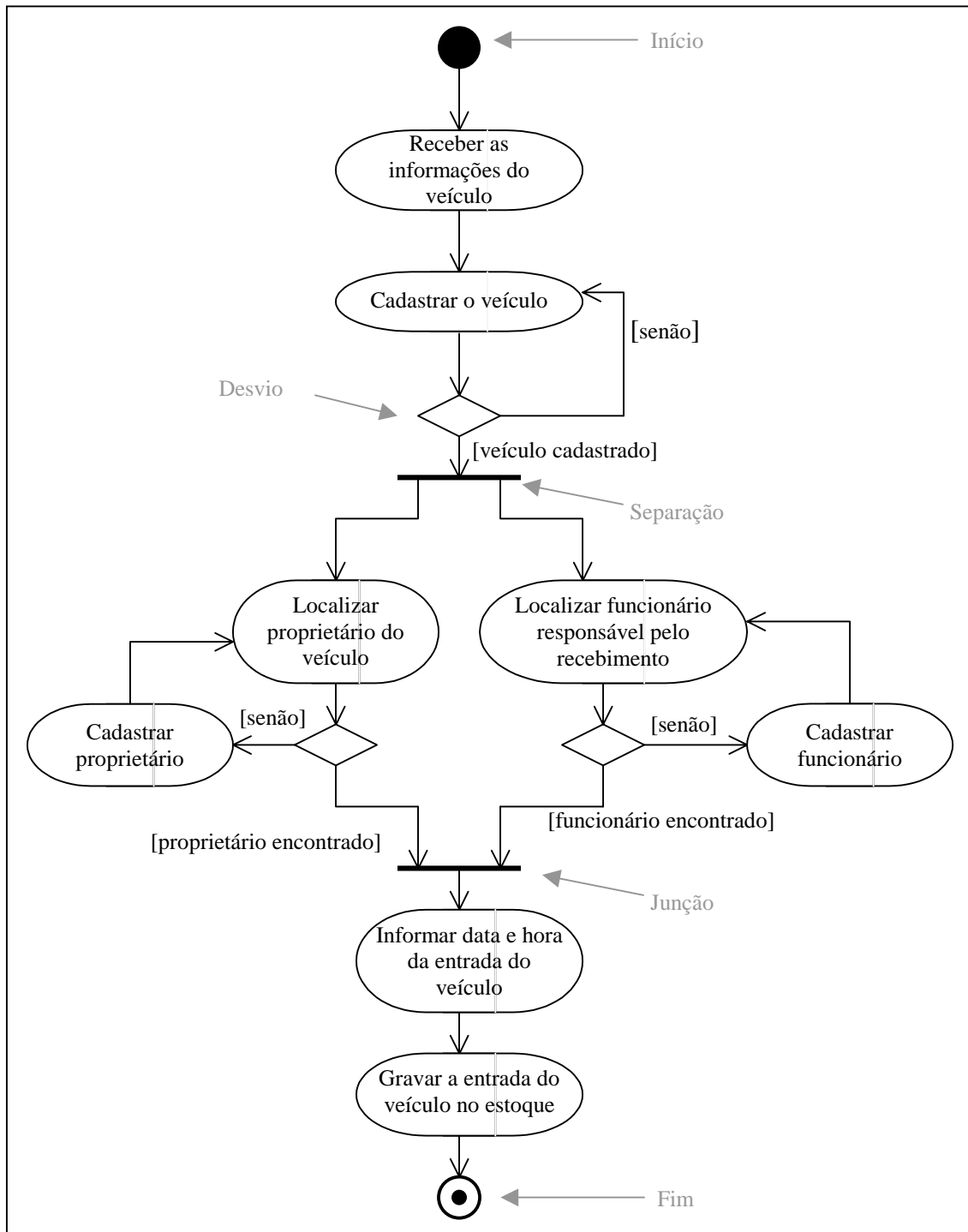


Figura 2.17: Diagrama de atividades para a Entrada de veículos no estoque

### 2.5.11 Diagrama Físico

É utilizado para demonstrar relacionamentos físicos de um sistema. Existem dois tipos: o diagrama de utilização e o diagrama de componentes. O primeiro demonstra como componentes de *software* e de *hardware* interagem. Cada nó deste diagrama indica uma unidade computacional. Sistemas distribuídos podem tirar vantagem do diagrama de utilização, pois ele fornece uma visão geral bastante interessante. Já no diagrama de componentes, está a representação de módulos físicos do código – freqüentemente colocados num pacote – e suas dependências (FOWLER, SCOTT, 2000).

Uma opção interessante é combinar os dois diagramas em um só, onde diagramas de componentes são colocados dentro dos nós do diagrama de utilização, assim é possível visualizar quais componentes trabalham em cada nó. Colocando o sistema exemplo adotado numa rede com conexão TCP/IP, pode-se verificar a aplicação do diagrama físico na Figura 2.14.

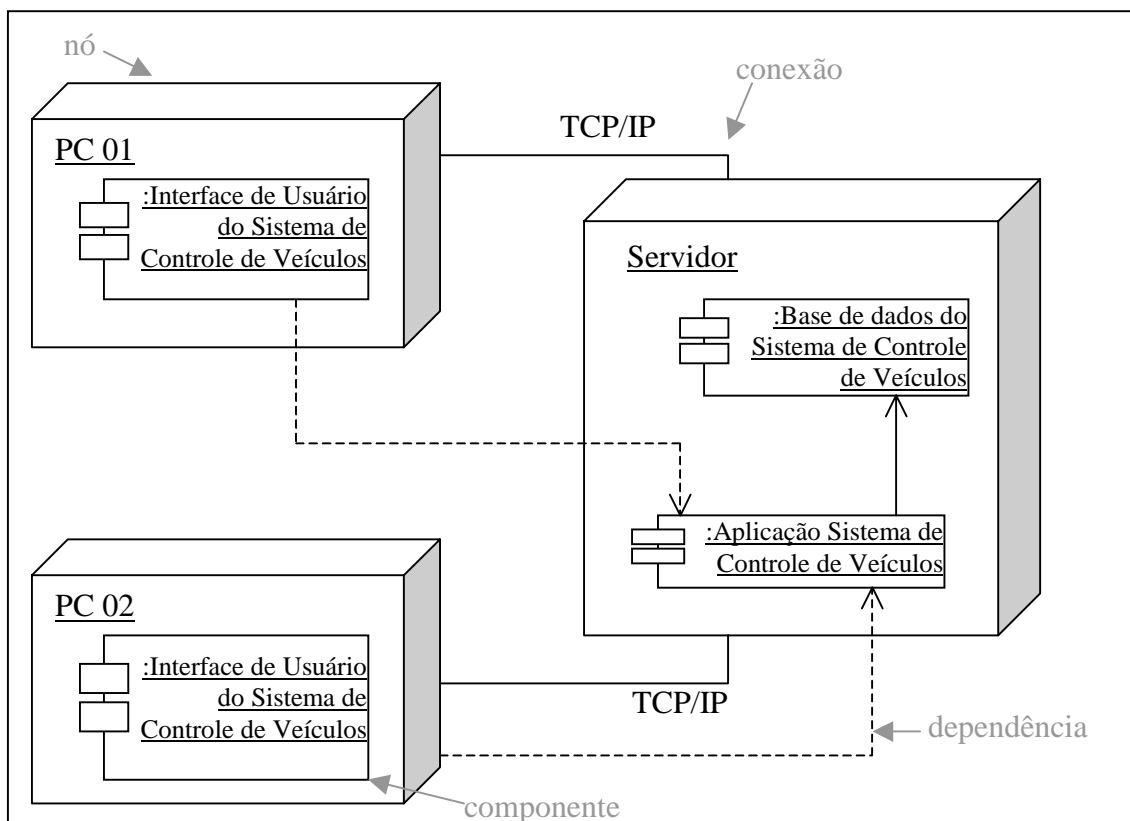


Figura 2.18: Diagrama de Componentes combinado com o Diagrama de Utilização

## 2.6 Agentes de Software

Outra tecnologia que pode trazer vantagens à utilização de um *site* é a de agentes de *software*. Faz parte da área de Inteligência Artificial, com base em entidades computacionais autônomas com capacidade de executar funções que beneficiem usuários ou outros agentes (COSER, 1999).

Um agente realiza suas atividades como se fosse uma espécie de auxiliar de uma pessoa, que por sua vez, pode preocupar-se com outros afazeres onde sua presença se faz mais necessária enquanto espera o resultado da missão incumbida ao agente.

É possível citar algumas das diversas aplicações que podem contar com a atuação direta e significativa de agentes de *software*, são elas: computação móvel, controle de processos industriais, administração de redes e sistemas, comércio eletrônico, correio eletrônico, interfaces adaptativas ou inteligentes e gerenciamento de informações pessoais, entre outros.

Coser (COSER, 1999) cita Dimitris Chorafas relatando que “a diferença chave entre agentes e as ferramentas atualmente disponíveis, é que o agente observa e se comunica com o usuário, enquanto uma ferramenta tradicional, como um objeto passivo, é algo que o usuário enxerga e manipula”.

### 2.6.1 Definição

Não existe um consenso quanto à definição de agentes, no entanto, vários pesquisadores conceituam agentes de acordo com o propósito com que são criados. Na seqüência estão algumas das definições mais encontradas.

Segundo Souza (1996) apud Gilbert et al. (1996), “agentes inteligentes são entidades de software que realizam algum conjunto de operações em benefício do usuário ou de outro programa, utilizando certo grau de independência ou autonomia, e ao fazê-lo, emprega algum conhecimento ou representação dos objetivos ou preferências do usuário. Para os autores, agentes inteligentes podem ainda ser definidos em função de três dimensões: agência (autonomia), inteligência e mobilidade”.



“Agentes são sistemas computacionais residentes em ambientes dinâmicos complexos, os quais percebem e atuam autonomamente, e ao fazê-lo, realizam um conjunto de objetivos e tarefas para os quais foram designados” (FARACO, 1998 apud SOUZA, 1996).

“Coen [COE 94] considera os agentes como entidades autônomas, capazes de agir em favor dos interesses dos usuários. Agentes devem ser robustos e capazes de manipular informações particulares com segurança. Agentes tendem a ser altamente interativos, participando ativamente do seu universo computacional, reagindo às alterações e provocando mudanças neste universo” (COSER, 1999 apud COEN, 1994).

Existem outras definições, mas variam basicamente de acordo com as propriedades que apresentam para realizar tarefas específicas.

## 2.6.2 Propriedades

A grande maioria dos agentes apresenta características como mobilidade, autonomia e inteligência. Mas, existem outras que são aplicadas de acordo com o tipo de função que realizam (SOUZA, 1996) (COSER, 1999) (SAVIO, 2000):

- Mobilidade – mover-se de uma máquina para outra, enquanto o estado interno é preservado;
- Autonomia – perseguir uma agenda sem a intervenção direta do seu usuário, requerendo aspectos de ação periódica, execução espontânea e iniciativa, em que o agente precisa ser capaz de tomar ações preemptivas (optativas) e independentes que eventualmente beneficiarão o usuário;
- Comunicabilidade – interação entre agentes com a intenção de atingir um objetivo em comum;
- Representatividade – representa o usuário (suas crenças, desejos, atributos e propósitos) através de suas ações;
- Veracidade – deve haver um alto grau de confiança de que o agente fornece informações verídicas e confiáveis;
- Degradação gradual – o agente deve completar sua missão ainda que alguma anomalia ocorra, por exemplo, se o sistema que está consultando

tiver problemas, o agente migra para outro e segue em busca do seu objetivo;

- Benevolência – garantia de que não haja objetivos conflitantes;
- Racionalidade – o agente atua para concluir suas metas e não de forma que impeça a realização dos objetivos;
- Persistência – mantém um estado interno conciso através do tempo, sem alterá-lo ao acaso;
- Planejamento – o agente sintetiza e escolhe entre diferentes cursos de ações, com o propósito de alcançar seus objetivos, por exemplo, quando migra de uma máquina para outra, se percebe que o caminho está obstruído ou com dificuldades, escolhe outra forma de chegar ao seu objetivo;
- Pró-atividade – Não respondem somente ao ambiente, mas perseguem um objetivo;
- Reatividade – Atuam de acordo com a percepção de alterações no ambiente em que estão inseridos e respondem prontamente a elas;
- Inteligência – capacidade do agente em desempenhar suas funções ou negociações quando se defronta com algo ambíguo. De acordo com a percepção do meio em que atua deve ser capaz de tomar decisões com relação à sua próxima atividade;
- Aprendizagem – o agente é capaz de adquirir sozinho novos conhecimentos com base no que já sabe.

Existem outras propriedades que são aplicadas de acordo com a função determinada aos agentes. As descritas aqui já são suficientes para caracterizar os agentes de *software* propostos nesse trabalho.

### 2.6.3 A Sociedade de Agentes

A atuação de vários agentes em prol de um único objetivo é encontrada em sistemas multi-agentes. Na seqüência, estão alguns aspectos que organizam a participação de agentes em um determinado trabalho, são eles (FARACO, 1998):

1. COOPERAÇÃO – trata do compartilhamento do conhecimento entre os agentes. Pode ser realizado de duas maneiras: *Task Sharing*, onde os agentes respondem a solicitação de outro agente que precisa de ajuda na realização de alguma tarefa; e *Result Sharing*, no qual os agentes disponibilizam suas informações considerando que algum outro possa precisar delas para desempenhar alguma função. Nesse modelo estão relacionadas algumas arquiteturas:
  - a. Arquitetura Quadro-Negro: a comunicação entre os agentes é realizada por meio de uma estrutura nomeada quadro-negro, onde perguntas e respostas são depositadas;
  - b. Arquitetura de Troca de Mensagens (*Message Passing*): os agentes comunicam-se diretamente entre si, mas pode haver a presença de um agente intermediário conhecido por agente facilitador;
  - c. Arquitetura Federativa (*Federal System*): quando há um grande número de agentes envolvidos num projeto, torna-se interessante dividi-los em grupos ou federações e é instituída a figura de um agente facilitador que ao receber alguma solicitação sabe se a mesma se refere a algum membro de seu grupo. Se for o caso, a mensagem é entregue diretamente para o destinatário, sem a necessidade de enviá-la para todos os componentes do grupo.
2. COORDENAÇÃO – refere-se à maneira com que os agentes são organizados. São dois os mecanismos abordados:
  - a. Mestre-Escravo: existem os agentes que coordenam e distribuem as tarefas e os agentes que realizam efetivamente as tarefas;
  - b. Mecanismo de mercado: todos se encontram no mesmo nível, onde cada um sabe o que os outros agentes são capazes de fazer. A vantagem há uma diminuição significativa na troca de mensagens.

3. PROTOCOLO DE COMUNICAÇÃO – para haver sucesso na comunicação dos membros da sociedade se faz necessária a instituição de uma linguagem comum de forma que todos possam se entender. O protocolo e linguagem de comunicação mais utilizados na atualidade estão contidos na ferramenta KQML (*Knowledge Query and Manipulation Language*).

#### 2.6.4 Classificação

A exemplo do que ocorre com a definição não há um consenso com relação à classificação ou tipo de agentes, no entanto é possível fazer uma distinção entre os diferentes tipos de agentes de acordo com as características em comum que apresentam (SAVIO, 2000) (FLEISCHHAUER, 1996):

- Agentes Colaborativos: cada agente tem sua especialidade, mas o grupo atua na solução de um único objetivo. As características principais são: autonomia e cooperação;
- Agentes de Interface (Aprendizes ou Adaptativos): atuam para facilitar a interação de usuários com *softwares* monitorando as atividades e assimilando os hábitos para serem sugeridos em utilizações posteriores. Os aspectos mais importantes são a autonomia e o aprendizado;
- Agentes Móveis: locomovem-se de um computador para outro através de redes de computadores em busca de informações solicitadas por seu proprietário. Têm os atributos de mobilidade, autonomia e cooperação;
- Agentes de Informação: realizam atividades de gerenciamento, manipulação e reunião de informações específicas localizadas em origens distribuídas. As principais propriedades são mobilidade, autonomia e cooperação;
- Agentes Reativos: suas atividades dependem da percepção do meio em que atuam. Quando há alguma alteração desempenham alguma função reagindo à mesma. A idéia é ter a participação de um grande número de entidades com estrutura simples para a obtenção de um comportamento complexo inteligente.

### 3 MODELO PROPOSTO

O Portal Pessoal Universal é um local na *Web* através do qual é possível localizar *Sites* Pessoais Universais partindo de informações de identificação de uma determinada pessoa, como CPF, RG, nome, data de nascimento, *Código Pessoal Universal*, e também, numa busca avançada, pode-se informar a descrição física de alguém para localizar seu *site* pessoal.

Algumas “entradas” podem resultar em mais de uma resposta, onde cada item da lista gerada contém a foto do titular do *Site* Pessoal, além de uma breve descrição com nome completo, data de nascimento, profissão, enfim informações que possam ser reveladas e que procurem identificar uma determinada pessoa.

Com isso, há o surgimento de uma nova classe de domínio, o “.pes”, onde todo *Site* Pessoal terá o formato www.nomefantasiadotitular.pes.país, podendo ser acessado diretamente com o endereço ou localizado através do Portal Pessoal Universal.

A proposta do *Site* Pessoal Universal é organizar as diversas informações relativas a um indivíduo, de uma forma padronizada e num único local, na forma de sessões. Elas poderão ser incluídas ou não de acordo com os critérios da pessoa que está construindo o seu *Site* Pessoal. Abaixo estão descritos brevemente exemplos de algumas delas, juntamente com a sugestão de conteúdo:

- Pessoal: contém a identidade universal com telefone, endereço, perfil físico, estado civil, etc;
- Profissional: armazena documentos como currículo, uma *home page* oferecendo os serviços profissionais aos clientes e outros;
- Saúde: abriga um *link* para o prontuário médico de um *site* especializado e também contém informações como tipo sanguíneo, seguro de saúde, preferências religiosas, exames, doenças anteriores, tratamentos atuais, alergias a medicamentos, produtos ou alimentos, etc;
- Tenho: é composta pelos itens que o titular do *site* deseja **vender**;
- Quero: constituída pelos itens que o titular do *site* deseja **comprar**.

Cada sessão é estruturada na forma de níveis de acesso para garantir a privacidade e segurança das informações ali contidas. São eles:

1. Público;
2. Emergencial;
3. Conhecidos e Colegas;
4. Amigos e Família;
5. Cônjuge; e,
6. O titular, de privacidade máxima.

A Fig. 3.1 esquematiza as sessões com os níveis de acesso, bem como exemplos de seus conteúdos.

Sessão \ Níveis de acesso	PESSOAL	PROFISSIONAL	MÉDICO	TENHO	QUERO
PÚBLICO	Identidade		Contato	Reserva made	Reserva seguradora
EMERGÊNCIA	Nome Contatos Colegas	Atestado Colegas de profissão	Médico Seguro Tipo sanguíneo Alergias		
CONHECIDOS/COLEGAS	Estado civil Horário	Situação/cargo Cartão de visitas Currículo Referências			Transporte solidário para o trabalho
AMIGOS/FAMÍLIA	Histórico escolar	Parcerias	Condições de saúde Prescrições	Passagem de viagem	Babá Espaço para festa particular
CÔNJUGE	Testamento pessoal			Deix de família	
O TITULAR				Presenças	

**Figura 3.1: Representação das Sessões e dos Níveis de Acesso, com sugestão de informações a serem armazenadas no Site Pessoal Universal.**

A inclusão de informações fica sob a responsabilidade do titular do *Site Pessoal* e é realizada através do preenchimento de um formulário específico para cada documento a ser disponibilizado.

Propõe-se a criação de um *software* Construtor de *Sites* Pessoais Universais (também referido como CSPU), para auxiliar a construção de um *Site* Pessoal Universal. O mesmo é disponibilizado no próprio Portal Pessoal Universal. Através dele é possível distribuir nas sessões diversos documentos como CPF, Identidade, carteira de motorista, certidão de nascimento, certidão de casamento, passaporte, currículo, identidade

universal (com endereço, telefone, perfil físico, estado civil, etc), testamento pessoal, informações de crédito, conta bancária, investimentos, impostos, prontuário médico (tipo sanguíneo, medicamentos, alergias, seguro de saúde, etc), advogado e outros. Para ter mais segurança, sugere-se a protocolização antecipada dos mesmos, onde receberão data e hora para que sejam garantidas sua existência e integridade a partir daquele momento.

### **3.1 Sessão Pessoal**

Nesta sessão encontra-se a Identidade Universal do titular, onde estão várias informações como nome completo, endereço, telefone, estado civil, perfil físico, além de contatos, carteira de motorista, carteirinhas de sócio de clubes, cartão de crédito/débito, etc. Assim, se acontecer de mudar, por exemplo, de endereço ou telefone, basta alterar os dados de interesse e todos os documentos que utilizam estas informações serão automaticamente atualizados.

Também contém um conjunto de diretórios *on-line* onde estão listados os nomes de pessoas conhecidas que, na verdade, são um *link* para os seus respectivos *Sites* Pessoais Universais. Isto ocorre da mesma forma que a identidade universal do titular do *site* em questão encontra-se vinculado ao seu nome nos diretórios *on-line* de pessoas conhecidas.

### **3.2 Sessão Profissional**

Este item apresenta uma descrição de como é possível utilizar a *Web* para procurar um emprego através da disponibilização do currículo on-line.

A construção do currículo é feita através da utilização do CSPU, ativando um assistente que disponibiliza uma série de módulos a serem preenchidos para identidade universal (a mesma da Sessão Pessoal), educação, experiência, prêmios, metas e outros dados pessoais relevantes.

O processo de procura de emprego conta com o auxílio de um *software* agente selecionador de oportunidades que preencham requisitos como tipos específicos de empregos, o porte da empresa, horário de trabalho, local, margem salarial, etc. O titular do *site* pode ativar seu agente e esperar que uma lista de potenciais empregos lhe seja apresentada para analisar qual é mais interessante e então entrar em contato com o possível empregador.

Uma outra forma, seria trabalhar com um campo de estado atual de disponibilidade de trabalho, cujo objetivo é demonstrar o quão interessado está um indivíduo em procurar um emprego, o titular do *site* altera o valor deste campo e espera que empresas encontrem o seu currículo, já que muitas delas adotam esse processo para recrutamento de profissionais, ou seja, utilizam um Agente Selecionador de Profissionais candidatos a um determinado cargo.

Os possíveis valores para este campo são:

1. Não está disponível, nem à procura de emprego;
2. Não está à procura de emprego, mas gostaria de ficar de olho no mercado de trabalho;
3. Interessado em ser convidado para um almoço;
4. Interessado em receber um telefonema de recrutamento;
5. Disponível para entrevistas imediatamente;
6. Vai considerar trabalho com contrato;
7. À procura de trabalho temporário, com contrato ou qualquer posição disponível.

Esta última maneira, torna a seleção de profissionais muito mais rápida por parte das empresas, pois as informações contidas nos currículos selecionados são atuais e corretas, portanto não há atrasos por conta de enganos de digitação em números de identificação ou localização de candidatos.

O responsável pelo recrutamento de pessoal também pode programar o Agente Selecionador de Profissionais para avisá-lo quando um profissional altamente desejável altere seu estado de disponibilidade profissional para “*interessado em ser convidado para um almoço*”.



Assim que um candidato é avisado da aceitação de seu currículo em uma empresa e aceita essa oferta, ele muda o campo de estado de disponibilidade de trabalho em seu currículo para 0, indicando que não está disponível, nem à procura de emprego. E, se por acaso estiver inscrito em listas de recrutamento de pessoal de outras empresas, automaticamente será removido das mesmas apenas com a alteração deste campo.

O currículo universal tem uma outra vantagem: ele abre espaço para a inclusão de um vídeo digital, onde o candidato pode apresentar-se brevemente informando seus objetivos, experiências, e possivelmente respondendo a um questionário previamente fornecido pela empresa. Assim, é possível analisar outros fatores do candidato como comportamento e perfil emocional, pois não basta ser o melhor profissional, ele precisa se adaptar ao estilo da empresa. O vídeo digital também se apresenta de grande vantagem para aqueles que têm dificuldade em ir até a empresa para fazer uma entrevista pessoal no horário marcado.

O currículo universal tem dois níveis de acesso:

- Público, sem a divulgação de dados pessoais; apenas ficam visíveis informações que indiquem o perfil profissional;
- Total, onde após analisar a proposta de emprego e aceitá-la, libera as demais informações contidas no currículo, transferindo-o na íntegra para o nível de acesso público.

O currículo universal também oferece a oportunidade de um profissional colocar-se no mercado de trabalho de uma forma rápida e eficiente. Além de poder analisar as várias propostas de trabalho e selecionar a que mais lhe interessa, num tempo muito menor ao do sistema que hoje é adotado.

Além do currículo, o titular do *site* pode colocar nesta sessão um *link* para o *Web Site* Profissional que já está desenvolvido, onde expõe seus serviços e capacidades profissionais.

### 3.3 Sessão Saúde

Nesta sessão está localizado um prontuário médico reduzido do titular do *site*. Nele estão contidas informações como tipo sanguíneo, alergias a medicamentos, produtos ou alimentos, seguro de saúde, doenças e tratamentos anteriores e atuais, exames médicos realizados, instruções sobre o que fazer no caso de ficar de alguma forma incapacitado, em coma ou morte, preferências religiosas, histórico de transfusões, instruções para contatar parentes e o médico de confiança, enfim tudo o que diz respeito à saúde do titular do *site*. Também há espaço para a inclusão de um *link* para um prontuário médico localizado em uma empresa especializada no ramo.

Todos os documentos médicos incluídos no *Site* Pessoal Universal de alguém possuem cópias de segurança, em poder dos médicos que solicitaram os exames, para desfazer qualquer possível engano que venha a ocorrer.

O grande benefício desta sessão está na disponibilização de informações sobre a saúde de uma pessoa em casos onde ela e outras pessoas presentes estejam impossibilitadas de fornecer dados médicos numa situação de emergência. O procedimento da equipe médica nestes casos, seria acessar o Portal Pessoal Universal, e através de algum dado de identificação disponível ou descrição física, localizar o *Site* Pessoal Universal do paciente em questão, e obter algumas informações básicas sobre a saúde do paciente.

A ficha médica de uma pessoa possui seis níveis de privacidade, como todos os outros documentos contidos num *Site* Pessoal. O titular do *site* é quem controla as permissões de acesso a cada nível. Portanto, a equipe médica num primeiro momento pode visualizar somente as informações que o paciente decidiu tornar público previamente. Dentre essas informações encontra-se uma lista de nomes e dados de localização de várias pessoas a quem contatar nesses casos e que, de acordo com a situação, podem decidir quanta informação devem liberar.

O prontuário médico virtual pode ser de grande ajuda não só em casos de emergência, mas também quando uma pessoa precisa apresentar exames previamente realizados em uma nova consulta médica, quando ocorre transferência para outros

médicos ou para um outro hospital. Desta forma, basta acessar em seu *Site* Pessoal Universal, todos os documento relativos à sua saúde, atualizados, sem ter que percorrer inúmeros locais físicos para obtê-los. Para a realização do prontuário médico virtual há a “necessidade de uma normatização legal e ética que atenda às exigências contemporâneas, além do respeito e garantia dos princípios de segurança da informação: integridade, confidencialidade, disponibilidade e legalidade” (Revista MEDICINA Conselho Federal, 2001).

O preenchimento da ficha médica fica a encargo do titular do *site*, porém sob a supervisão e/ou responsabilidade do médico que acompanhou um determinado tratamento. Isto porque as informações contidas nesta ficha são destinadas a outros médicos, portanto devem ser expressas em termos técnicos específicos da área da saúde.

Na área da saúde, a utilização da Internet justifica-se não só pelo grande volume de informação que a prática da medicina e áreas afins manipula, como por possibilitar o acesso e o tratamento destas informações de forma rápida e segura. Há muito tempo, os serviços de saúde procuram formas confiáveis e privadas de armazenar os prontuários médicos, fato que constitui verdadeiro transtorno nas unidades, em função do espaço que ocupam. Algumas informações precisam permanecer preservadas por muito tempo, por razões epidemiológicas ou mesmo legais (Revista MEDICINA Conselho Federal, 2001).

### 3.4 Sessões de Demanda e de Oferta

Nessas duas sessões especiais, denominadas Tenho e Quero, uma pessoa encontra à sua disposição inúmeros itens previamente descritos em suas características, ou seja, um produto ou serviço que possui todos os seus detalhes discriminados numa estrutura conhecida como **Descritor Universal**<sup>1</sup>, que podem ser destinados à compra, quando colocados na Sessão Quero, ou à venda, quando colocados na Sessão Tenho.

No caso de não encontrar o descritor ou característica de um produto que deseje comprar ou vender, o titular pode enviar um pedido de inclusão do mesmo, juntamente

---

<sup>1</sup> Um Descritor Universal é uma ferramenta de descrição eletrônica de todas as características de um item que utiliza termos padronizados, inclusive aspectos da fabricação industrial de tal item.

com os atributos de interesse, aos administradores do Portal Pessoal Universal e tê-lo disponível em pouco tempo.

As atividades de compra e venda se dão por meio de Agentes de *Software*, denominados, respectivamente, **Agente de Compra** e **Agente de Venda**, presentes em cada *Site* Pessoal Universal. Eles fazem todo o serviço exaustivo de procurar um item na grande rede, seja para comprar ou para encontrar alguém que queira o produto que se deseja vender. Para realizar a pesquisa nos diversos servidores da Internet que possuem *Sites* Pessoais hospedados, criou-se a figura de dois Agentes de *Software*, únicos em cada servidor. São eles:

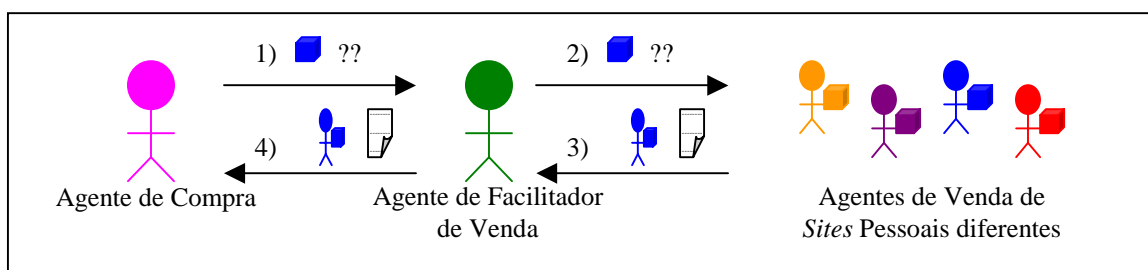
- **Agente Facilitador de Venda**, cuja função é facilitar a comunicação entre o Agente de Compra que chega ao provedor em busca de itens a serem adquiridos e os Agentes de Venda de cada *Site* Pessoal (Fig. 3.2); e,
- **Agente Facilitador de Compra**, com a missão de intermediar a “conversa” do Agente de Venda que chega ao provedor procurando alguém que queira comprar o que ele tem para vender e os Agentes de Compra de cada *Site* Pessoal do servidor pesquisado.

Na Fig.3.2, observam-se as etapas do processo de compra de algum produto com o auxílio de Agentes Facilitadores: 1) O Agente de Compra pergunta ao Agente Facilitador de Venda se existe alguém que queira vender o produto que ele quer comprar. 2) Este, por sua vez, transmite a solicitação aos Agentes de Venda de cada *Site* Pessoal daquele provedor. 3) Aquele que possui o produto procurado repassa suas informações de contato e os detalhes do produto em questão ao Agente Facilitador de Venda 4) que os entrega ao Agente de Compra que relata ao seu proprietário ao final da busca. Quando se tratar de uma venda de produto o comportamento dos agentes se dará de forma semelhante à compra aqui demonstrada, apenas alterando os agentes da transação.

A presença dos Agentes Facilitadores é muito importante por dois motivos que influenciam significativamente a velocidade na obtenção das informações procuradas: o primeiro, é que quando um *Site* Pessoal é criado, seu endereço é automaticamente incluído nos registros dos Agentes Facilitadores e, portanto, como sabem onde estão localizados os *Sites* Pessoais hospedados no servidor onde atuam, são muito rápidos

para encontrar o que procuram (afinal, num servidor existem muitos outros *sites* armazenados além dos *Sites* Pessoais). A outra razão está relacionada à segurança do servidor, onde os Agentes Facilitadores são entidades internas, com funções conhecidas e permissão já concedida para desempenhá-las, evitando assim a entrada de “estranhos”, onde seriam necessárias a solicitação e a concessão de acesso ao servidor.

Dessa forma, com a localização rápida do item procurado, os Agentes Facilitadores repassam as informações de contato e do produto aos Agentes externos que, por sua vez, armazenam as mesmas num relatório que é entregue ao final da busca à pessoa que os acionou. De posse das informações recebidas, o titular do *Site* Pessoal entra em contato com o dono do item que lhe interessa e realiza a transação da forma que lhe convier.



**Figura 3.2: Processo de compra e venda com Agentes Facilitadores**

A utilização de agentes facilita imensamente a procura por produtos ou serviços que se deseja vender ou adquirir, pois com a existência dos descritores é possível encontrar com maior precisão o que se está procurando, além de economizar tempo na cansativa procura através dos *sites* de busca, ou ainda, ter que disponibilizá-los em diversos *sites* de leilão eletrônico para obter um retorno mais rápido.

A seguir estão detalhados o comportamento, as atividades e necessidades de cada agente de *software*, dentro de suas respectivas sessões.

### 3.4.1 Agente de Compra – Sessão de Demanda

Quando uma pessoa deseja *comprar* algum item, ela preenche um formulário de busca que usa o descritor do item procurado com a indicação das devidas características, incluindo detalhes sobre as condições da compra, como variações de preço e condições de pagamento aceitáveis, além de disponibilizar um campo para indicar a extensão da

pesquisa a ser realizada pelo agente, ou seja, o número de respostas que espera. O resultado de uma pesquisa é automaticamente desconsiderado em buscas subsequentes com base no identificador único daqueles já encontrados.

O próximo passo, é ativar o **Agente de Compra** que procura nos *sites* do formato www.nomefantasiadotitular.pes.país (ou seja, em *Sites Pessoais Universais*), a Sessão TENHO para comparar o descritor do produto que tem consigo e o que está disponível para venda<sup>1</sup> no *site* localizado. No caso de uma comparação positiva (igualitária, pelo menos, na característica **nome**), o item e informações de seu proprietário são anexados à lista que será entregue ao dono do Agente de Compra, que por sua vez apenas aguarda pelo relatório com os possíveis itens a serem comprados. Feita uma análise das ofertas, o titular do *site* entra em contato com o proprietário do item escolhido e efetiva a compra da forma que os dois acordarem.

Esta sessão na prática é refletida em uma tabela chamada QUERO, onde o proprietário do *Site* Pessoal coloca itens que deseja COMPRAR. De forma mais detalhada, nesta tabela são disponibilizados os descritores dos objetos que o dono do *site* deseja adquirir.

É possível entender melhor esse processo observando os seguintes critérios do Agente de Compra:

1. Busca por um OBJETO, com algumas<sup>2</sup> características relatadas do respectivo descritor, como **nome** e **tipo**. Por exemplo, uma bicicleta do tipo corrida;
2. Possui um ESCOPO de procura, informado pelo usuário, tal como, uma cidade, um ou mais estados, um ou mais países; e,
3. Tem suas atividades encerradas de acordo com um NÚMERO DE RESPOSTAS, definido por seu proprietário. Por exemplo, os primeiros 20 itens encontrados.

Analisando o comportamento do Agente de Compra verifica-se alguns requisitos e a atuação do **Agente Facilitador de Venda**. São eles:

---

<sup>1</sup> Ver estado de mercado para venda no item 3.4.2 Agente de Venda – Sessão Tenho.

<sup>2</sup> Apenas a característica **nome** é obrigatória no preenchimento do formulário de busca de qualquer item.

1. Necessita de um mapa de servidores que possuem *Sites* Pessoais hospedados, com o qual definirá o escopo de procura a ser escolhido pelo usuário (SC, SC/PR/RS, Brasil, Brasil/Argentina, Mundo, etc.), a fim de saber onde buscar as informações;
2. Ao chegar num servidor, o agente de Compra “conversa” com o Agente Facilitador de Venda daquele servidor e pergunta se existe algum objeto que coincida com aquele que está procurando e que esteja disponível para venda;
3. O Agente Facilitador de Venda, por sua vez, que conhece o endereço de todos os Agentes de Venda daquele servidor, distribui a solicitação e cada Agente de Venda faz sua busca, nos seus respectivos arquivos (tabelas) **TENHO**, remetendo ao Agente Facilitador de Venda suas respostas; e,
4. O Agente Facilitador de Venda informa ao Agente de Compra o relatório que lhe foi repassado. Este, então, armazena o documento e segue para o próximo servidor.

Utilizando a linguagem XSQL (SQL estendida) num banco de dados Relacional-Objeto, pode-se verificar no Quadro 3.1 abaixo a criação da tabela **TENHO**, bem como uma possível consulta que procura por um objeto cujo nome seja “bicicleta”, de acordo com o exposto na sessão 2.4.2, especialmente o item 2.4.2.5:

```
create table bicicleta
(identificador integer,
nome char varying,
tipo char varying,
numMarcha integer,
cor char varying,
tempoUso char varying,
situação char varying,
fabricante char varying,
preço real,
estadoDeMercadoParaVenda char varying)
```

```
create table carro
(identificador integer,
nome char varying,
cor char varying,
listaAcessórios setof(minhaSeqüência),
fabricante char varying,
anoModelo char varying,
preço char varying,
estadoDeMercadoParaVenda char varying)
```

Observe a definição do tipo minhaSeqüência na Listagem 2.1, página 43.

↑  
char varying  
é o tipo seqüência variável de caracteres.

```
create table TENHO
(prodoto1 ref(bicicleta),
prodoto2 ref(carro))
```

↑ O item `prodoto1` faz referência ao objeto `bicicleta`, da mesma forma que o item `prodoto2` é uma referência ao objeto `carro`, ambos descritos acima e representados na forma de tabelas.

```
select TENHO.*.nome
from TENHO
where TENHO.*.nome = "bicicleta"
```

↑ O `*` indica que a busca será feita em todos os itens da tabela `TENHO`, neste caso, `prodoto1` e `prodoto2`, considerando o atributo `nome = "bicicleta"` de todos os itens.

**Tabela TENHO**

<i>Produto1 (ref(bicicleta))</i>	<i>Produto2 (ref(carro))</i>
{00001; bicicleta; corrida; 15 marchas; verde; 2 anos de uso; bom estado; caloi; R\$ 250,00}	{00001; gol; branco; (ar condicionado, trio elétrico, alarme), Volkswagen, 2000/2000, R\$ 18.000,00}

**Quadro 3.1: Demonstração de criação da tabela TENHO e de uma consulta na linguagem XSQL feita por um Agente de Compra em busca de uma bicicleta.**

### 3.4.2 Agente de Venda – Sessão de Oferta

Quando uma pessoa deseja *vender* algum item, ela coloca o respectivo descritor na Sessão TENHO de seu *Site* Pessoal Universal. O titular do *site* tem duas opções para realizar essa venda: ele pode agir passivamente, esperando que o Agente de Compra de outra pessoa encontre o seu produto, ou de maneira ativa, acionando o próprio Agente de Venda que sai pela Internet pesquisando em servidores que hospedam *Sites* Pessoais, alguém que queira comprar o que ele deseja vender.



Esperando pelo Agente de Compra de outras pessoas, o proprietário de um *Site* Pessoal faz uso do **estado de mercado para venda**, que indica se um determinado item está ou não à disposição e em que condições. Os valores possíveis são:

- “0” para “Não está à venda”;
- “1” para “Venderia ou aceitaria ofertas em circunstâncias especiais”;
- “2” para “O preço está fixado. A primeira pessoa que pagar, leva”;
- “3” para “Faça uma oferta”;
- “4” para “Preciso vender imediatamente”;
- “5” para “Maior oferta até certa data, leva”.

Portanto, alterar o valor do estado de mercado para venda de “0” para qualquer outro indica que o produto está à venda e o Agente de Compra de outra pessoa passa a considerar esse item ao pesquisar o *Site* Pessoal em questão. Mas, se o valor do estado de mercado para venda for “0”, o Agente de Compra ignora esse produto, em nem mesmo gasta seu tempo comparando as outras características do descritor. Depois de estabelecidos estes critérios, o passo seguinte é esperar pelas ofertas e realizar o negócio.

Mas, se o dono de um *Site* Pessoal tiver pressa em vender algo, não espera que alguém o encontre e, então aciona seu **Agente de Venda** que sai pela rede em busca de alguma pessoa que queira comprar o item que ele deseja se desfazer. O Agente procura em servidores que hospedam *Sites* Pessoais, os endereços do formato [www.nomefantasiadotitular.pes.país](http://www.nomefantasiadotitular.pes.país), pesquisa a Sessão QUERO do mesmo e verifica se os descritores são compatíveis, ou seja, se o descritor do item à venda que carrega consigo tem as mesmas características do item colocado nessa sessão. Uma vez realizada uma comparação positiva, o item, juntamente com informações de seu proprietário, é anexado à lista que será fornecida ao dono do Agente de Venda.

Esta sessão está fisicamente ligada a uma tabela denominada TENHO, onde o proprietário do *Site* Pessoal coloca produtos que deseja VENDER. De forma mais detalhada, nesta tabela são disponibilizados os descritores dos objetos cujo dono do *site* quer se desfazer. A seguir estão listados alguns requisitos necessários ao funcionamento do Agente de Venda:

1. Busca por um OBJETO localizado na sessão QUERO de outros *Sites* Pessoais, cujo nome coincida com aquele que se quer vender. Além do atributo nome (obrigatório), também são informados todos os aspectos do item para facilitar uma busca mais precisa, por exemplo, nome: bicicleta, tipo: corrida; constando todas as outras características, inclusive as mais específicas, como dimensões físicas;
2. Possui um ESCOPO de procura, informado pelo usuário, tal como, uma cidade, um ou mais estados, um ou mais países;
3. Tem suas atividades encerradas de acordo com um NÚMERO DE RESPOSTAS, definido por seu proprietário. Por exemplo, os primeiros 50 itens encontrados, que podem ser desconsiderados, através de seus identificadores únicos, em consultas subseqüentes.

Com relação ao comportamento do Agente de Venda verifica-se algumas necessidades e a atuação do **Agente Facilitador de Compra**. São eles:

1. Necessita de um mapa de servidores que hospedam *Sites* Pessoais, o qual definirá o escopo de procura a ser escolhido pelo usuário (SC, SC/PR/RS, Brasil, Brasil/Argentina, Mundo, etc.), a fim de saber onde buscar as informações;
2. Ao chegar num servidor, o agente de Venda “conversa” com o Agente Facilitador de Compra daquele servidor e pergunta se existe algum objeto que coincida com aquele que está procurando;
3. O Agente Facilitador de Compra, por sua vez, que conhece o endereço de todos os Agentes de Compra daquele provedor, distribui a solicitação e cada Agente de Compra faz sua busca, nos seus respectivos arquivos (tabelas) **QUERO**, remetendo ao Agente Facilitador de Compra suas respostas;
4. O Agente Facilitador de Compra informa ao Agente de Venda o relatório que lhe foi repassado. Este, então, armazena o relatório e segue para o próximo servidor de seu mapa de pesquisa.

Os Agentes de Compra e de Venda devem apresentar as seguintes características, cujos conceitos estão relatados no subcapítulo 2.6 (SOUZA, 1996): mobilidade, autonomia, comunicabilidade, representatividade, veracidade, degradação gradual, benevolência, racionalidade, persistência, planejamento e pró-atividade.

O que se observa neste modelo, é uma clara distinção entre informações que precisam estar sob um rígido esquema de segurança e outras que não necessitam de tal cuidado. Nas Sessões Pessoal, Profissional e Saúde, informações sigilosas e estritamente particulares, são organizadas. A distribuição de tais informações nos diversos níveis de acesso – *Público (1)*, *Emergencial (2)*, *Conhecidos e Colegas (3)*, *Amigos e Família (4)*, *Cônjuge (5)*, e *O Titular (6)* – ficam ao encargo do proprietário do *site*. Como se trata de informações de altíssimo valor, a segurança deve ser planejada e executada com muito critério e precisão.

Certificados Digitais são uma opção interessante na atualidade para garantir, em primeiro lugar, que um *Site Pessoal* seja criado somente pelo proprietário dos dados que serão incluídos, ou seja, uma pessoa não pode criar um *Site Pessoal* com os dados de outra. Num segundo momento, certificados digitais podem controlar perfeitamente que somente determinadas pessoas tenham acesso às sessões, ato que não é possível ser garantido apenas com o uso de senhas, pois podem ser descobertas com mais facilidade, além de não haver o vínculo senha-pessoa.

Como nesta proposta existem seis diferentes níveis de acesso, são necessários seis Certificados Digitais para garantir a privacidade e distinção de acesso às informações contidas nestas sessões. Em função disso e devido ao alto grau de complexidade envolvido para garantir a segurança de informações tão preciosas, optou-se por especificar nesta dissertação um **modelo genérico de Site Pessoal**. No que diz respeito à segurança sugere-se a utilização de seis certificados digitais: um com permissão para leitura e escrita que é exclusivo do titular do *site* – conhecido por Código Pessoal Universal, e os outros cinco, com permissão de leitura de algumas informações conforme o nível associado, apesar de apresentar-se um tanto inconveniente. O projeto e a implantação de acesso controlado em níveis se apresentam mais complexos do que aparentam, por isso a segurança do *Site Pessoal* fica apenas indicada aqui e seu estudo aprofundado merece exclusividade em trabalhos futuros.

No próximo capítulo, é apresentada a proposta de um Portal Pessoal Universal, através do qual é possível organizar, consultar e manipular vários *Sites* Pessoais Universais desenvolvidos de acordo com o modelo descrito neste capítulo.

## 4 O PORTAL PESSOAL UNIVERSAL

O Portal Pessoal Universal, cujo modelo está representado na Fig. 4.1, é um local na Internet, que centraliza *Sites* Pessoais Universais. É através dele que se pode localizar os mesmos utilizando algumas “palavras-chave”, como nome, o Código Pessoal Universal, CPF, Identidade, data de nascimento, e também, em uma busca avançada, pode-se informar a descrição física de alguém para localizar seu *Site* Pessoal, contando com os seguintes parâmetros de distinção e opções de escolha:

- Cor dos cabelos: preto, castanho escuro, castanho claro, loiro, ruivo, branco e outra (informar a cor predominante);
- Tipo dos cabelos: lisos, encaracolados, crespos;
- Tamanho dos cabelos: sem cabelo, pouco cabelo, curto, médio, comprido;
- Cor da pele: branca, negra;
- Cor dos olhos: preto, castanho escuro, castanho claro, verde, azul, acinzentado;
- Biotipo: magro, médio, obeso;
- Altura: até 1,50m, entre 1,50m e 1,70m, entre 1,70m e 1,90m, acima de 1,90m;
- Marca de identificação: sinal (informar), cicatriz (informar), marca de nascença (informar), tatuagem (informar), outra (informar), ausente;
- Idade aproximada: (informar);
- Deficiência física: (informar).

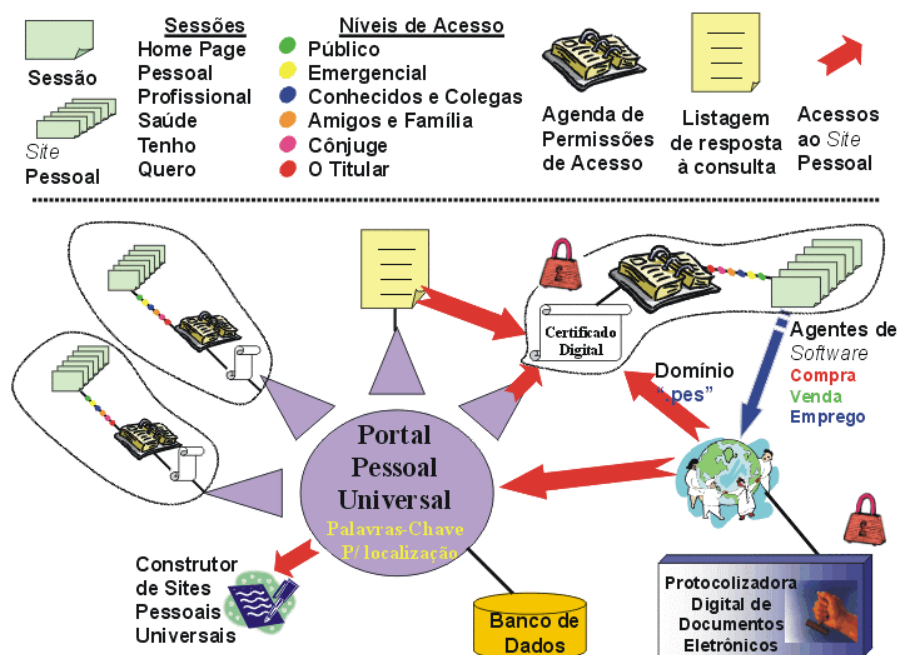


Figura 4.1: O modelo do Portal Pessoal Universal.

A navegação fica restrita à página inicial, num primeiro momento, por seu conteúdo ser de domínio público. O acesso ao restante do *site*, ou seja, às informações que estão inseridas nas sessões, fica limitado àquelas pessoas que possuem permissão de acordo com os seis níveis de acesso definidos anteriormente – público, emergencial, conhecidos e colegas, amigos e família, cônjuge e o titular.

Para modelar o funcionamento do Portal Pessoal Universal como um todo, optou-se pela utilização da linguagem de modelagem de sistemas UML – *Unified Modeling Language*, por demonstrar e documentar com exatidão, profissionalismo e clareza o projeto de um sistema. À medida que os diagramas UML são construídos, a equipe de trabalho envolvida participa de todas as etapas de desenvolvimento, tendo assim, conhecimento mais aprimorado do sistema por inteiro, ao invés de apenas partes dele. Além disso, esta linguagem de modelagem apresenta as etapas de um sistema de forma que analistas, programadores e clientes podem entender e discutir propostas de funcionamento de maneira mais produtiva e precisa. Os detalhes dessa linguagem podem ser verificados no capítulo 2 (no subitem 2.5).

Desta forma, na seqüência estão expostos os diagramas de atividade e de casos de uso que organizam, documentam e especificam as informações das principais funções do Portal Pessoal Universal.

## DIAGRAMAS DE ATIVIDADES

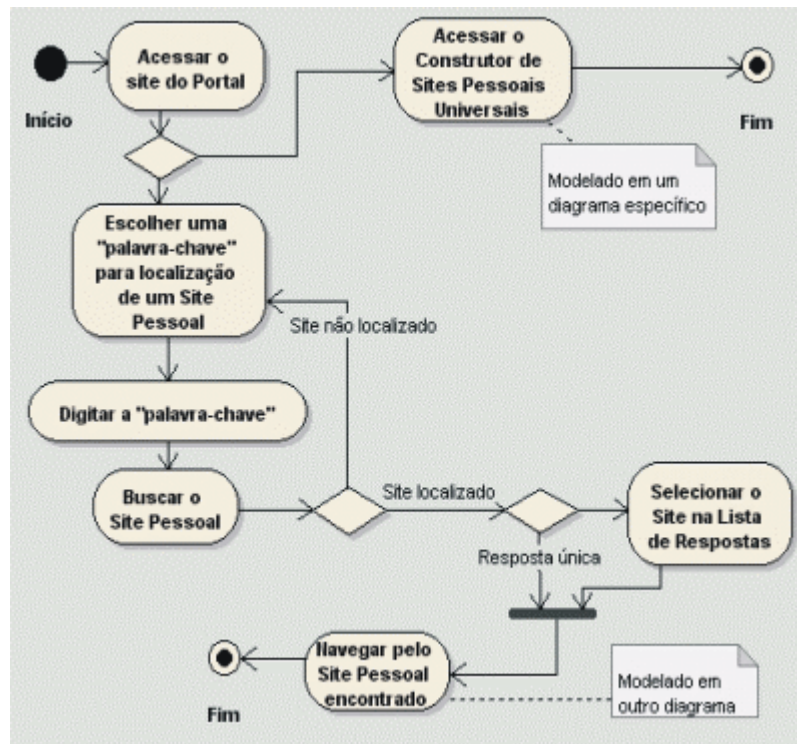


Figura 4.2: Diagrama de Atividades do Portal Pessoal Universal

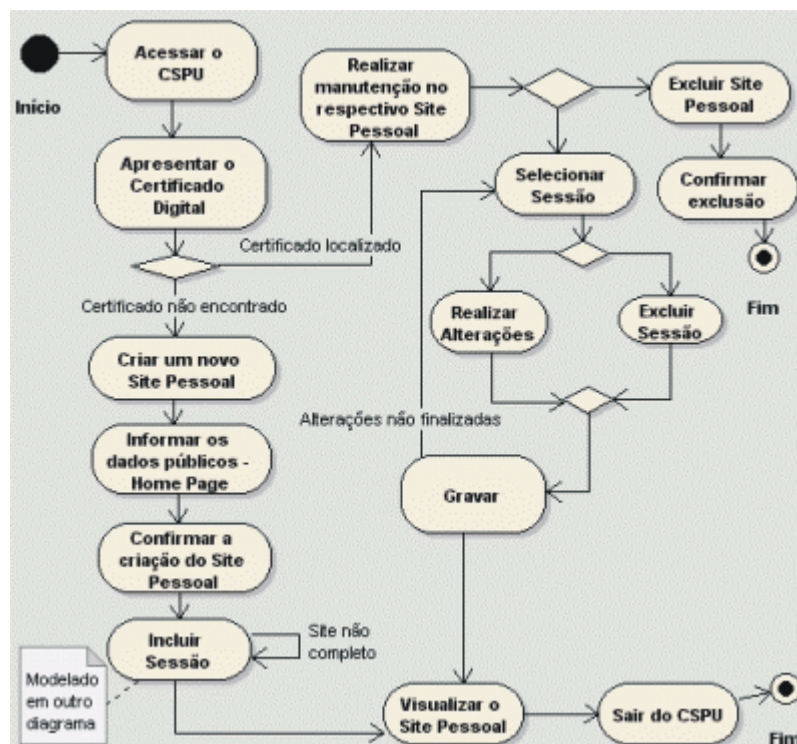


Figura 4.3: Diagrama de Atividades do Construtor de Sites Pessoais Universais - CSPU

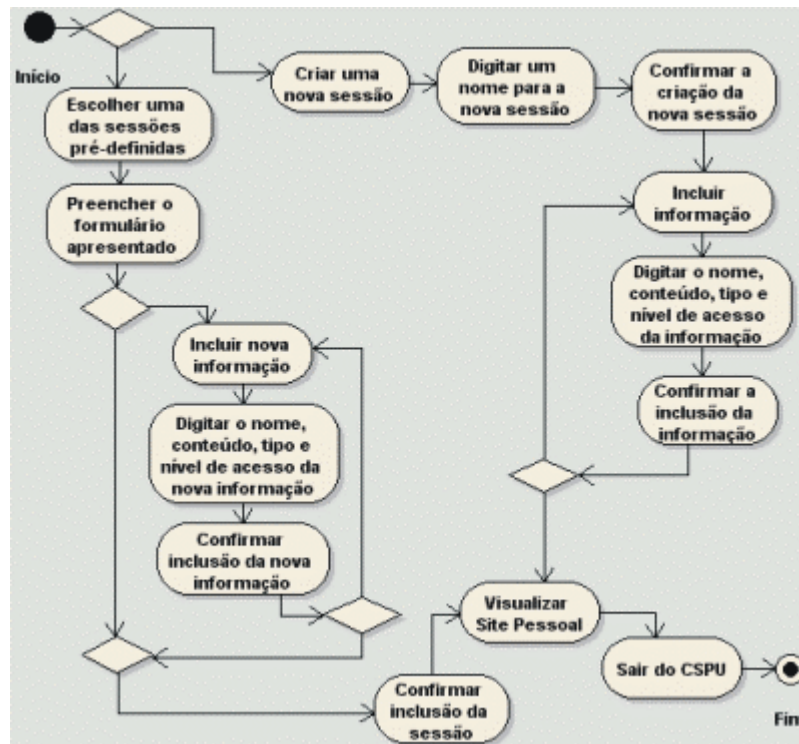


Figura 4.4: Diagrama de Atividades da Inclusão de Sessões

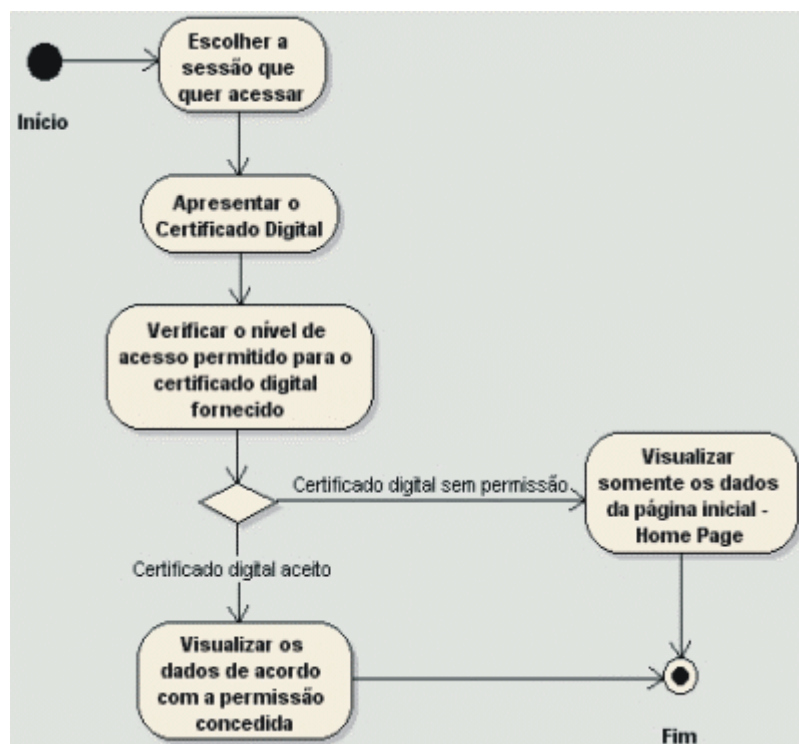


Figura 4.5: Diagrama de Atividades da Navegação em um Site Pessoal



## CASOS DE USO (Uses Case)

### Caso de Uso – Localização de um *Site* Pessoal

Descrição: um internauta acessa o *site* do Portal Pessoal Universal, escolhe uma das “palavras-chaves”, digita o conteúdo dela e dá início à localização do *Site* Pessoal que está procurando.

Pré-condição: escolher e digitar o conteúdo de uma “palavra-chave” que serve como parâmetro para a localização de um *Site* Pessoal.

Resultado: se houver mais que uma ocorrência, tem-se uma lista com os *Sites* Pessoais encontrados, em função do parâmetro de busca inicial. Caso contrário, o internauta é remetido diretamente à página inicial do *Site* Pessoal procurado.

Ator: um internauta

Passos:

- acessar o Portal
- escolher a palavra-chave
- digitar a palavra-chave
- iniciar a busca pelo *Site* Pessoal

### Caso de Uso – Criação de um *Site* Pessoal

Descrição: um internauta inicializa o Construtor de *Sites* Pessoais Universais, o servidor de *Internet* solicita o certificado digital, o mesmo é apresentado e não é encontrado. Portanto, o usuário digita seu nome, seleciona uma foto e um texto descritivo ou de “Boas Vindas”, confirma a criação do *Site* Pessoal e o vínculo entre o *Site* e o certificado digital apresentado é registrado.

Pré-condição: possuir um certificado digital.

Resultado: um *Site* Pessoal com apenas a página inicial, a *Home Page*.

Ator: Titular do *Site* Pessoal.

Passos:

- iniciar o Construtor de *Sites* Pessoais Universais
- apresentar o Certificado Digital
- digitar informações
- criar o *Site* Pessoal
- registrar o vínculo *Site* Pessoal-Certificado digital

### Caso de Uso – Inclusão de Sessões Pré-definidas

Descrição: o Titular do *Site* Pessoal escolhe uma das sessões pré-definidas, o formulário específico é apresentado, as informações sugeridas são preenchidas de acordo com o critério do Titular. A sessão é inserida no *Site* Pessoal. Outras informações podem ser incluídas nas sessões.

Pré-condição: um *Site* Pessoal criado.

Resultado: um *Site* Pessoal com uma ou mais sessões.

Ator: Titular do *Site* Pessoal.

Passos:

- escolher uma sessão pré-definida
- preencher o formulário específico
- inserir sessão no *Site* Pessoal
- incluir novas informações na sessão escolhida

### Caso de Uso – Navegação em um *Site* Pessoal

Descrição: um internauta escolhe a sessão que quer visitar, apresenta seu Certificado digital, se tiver permissão para acessá-la, visualiza as informações permitidas para seu nível de acesso.

Pré-condição: apresentar o Certificado digital

Resultado: visualização de sessão de acordo com o nível de acesso conferido ao visitante.

Ator: um internauta.

Passos:

- escolher uma sessão para ser visitada
- apresentar certificado digital
- visualizar informações de acordo com o nível de acesso do visitante

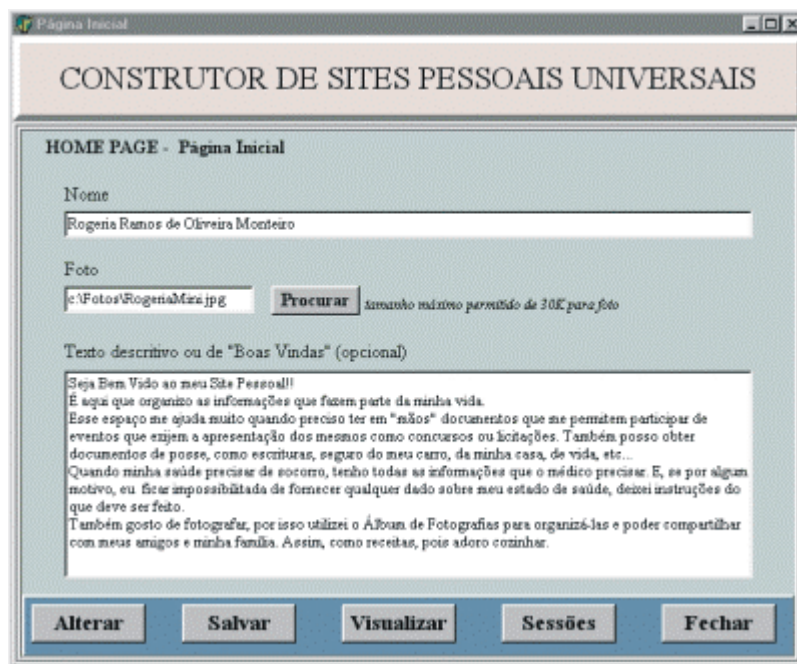
Antes de dar seqüência, faz-se necessário observar que a implementação do Portal não pertence ao escopo deste trabalho, portanto as janelas que serão encontradas durante a descrição do mesmo são apenas uma sugestão para a interface e tem como objetivo concretizar um pouco a idéia aqui proposta. Por questões de facilidade e domínio de ferramenta, optou-se por utilizar a linguagem de programação *Delphi* para demonstrar o funcionamento desta aplicação, mesmo sabendo que existem outras linguagens específicas e mais apropriadas para o desenvolvimento de *sites* para a Internet.

Portanto, abaixo está a descrição do Portal Pessoal Universal. Na Fig. 4.6, é possível observar a página inicial, com os parâmetros de busca e com a disponibilização do Construtor de *Sites* Pessoais Universais.

**Figura 4.6: Página inicial do Portal Pessoal Universal**

O Construtor de Sites Pessoais Universais – CSPU – é um *software* que guia uma pessoa a construir um *Site* Pessoal através de formulários específicos que organizam informações agrupando-as em sessões previamente definidas. Também permite incluir outras informações às sessões apresentadas e criar novas sessões que sejam de interesse do titular do *site*.

Na primeira etapa, Fig. 4.7, o titular fornece sua foto, seu nome completo e um breve texto a seu critério (opcional). Com essas informações, já é possível criar o *Site* Pessoal. O servidor que vai armazenar o *Site* Pessoal é programado para solicitar o certificado digital no momento da criação. O CSPU realiza então uma busca pelo certificado digital apresentado. Caso encontre, apresenta ao navegador o respectivo *Site* Pessoal, senão cria um novo *Site* Pessoal vinculado ao certificado digital fornecido. Nesse momento, ele é composto apenas pela página inicial.



**Figura 4.7: Elaboração da página inicial do *Site* Pessoal no Construtor de *Sites* Pessoais Universais**

Em seguida, o proprietário do *site* inicia a inclusão das sessões. O Construtor apresenta uma lista de sessões pré-definidas – Pessoal, Profissional, Saúde, Quero e Tenho – que, ao serem escolhidas apresentam ao usuário o respectivo formulário com as informações que constarão em tal sessão.

## 4.1 Sessão Pessoal

O conteúdo proposto para esta sessão (Fig. 4.8) é: nome, endereço, telefone, e-mail, assinatura manual digitalizada, sexo, data de nascimento, estado civil, contatos, descrição física, CPF, Identidade, Título de eleitor, Carteira de motorista (CNH), além de um álbum de fotografias, um caderno de músicas e um livro de receitas. Se o titular do *site* desejar acrescentar alguma informação nessa sessão que não seja uma das apresentadas, há um espaço onde pode solicitar a inclusão da mesma (Fig.4.9).

Inclusão de Sessões

## CONSTRUTOR DE SITES PESSOAIS UNIVERSAIS

**SESSÕES** Pessoal Quero uma sessão que não está disponível!

Pessoal

**SESSÃO PESSOAL** De acordo com a legenda ao lado, escolha o nível de acesso (NA), para cada informação. 1 - Público 2 - Essencial 3 - Conhecidos e Colegas 4 - Amigos e Família 5 - Contatos 6 - O Titular

Nome:  1

Endereço:  2

Telefone:  3

E-mail:  3

Nascimento:  4 Sexo:  M  F 1

Estado civil:  Casado  Solteiro  Divorciado  Viúvo 1

Contatos:

Nome:  1

Link:  1

Se quiser, acrescente ao nome, o telefone e o grau de relacionamento com a pessoa indicada. Por exemplo: João(mãe) (49) 2223433

Assinatura digitalizada: 4

**Perfil Físico** 1

**Documentos Pessoais** 4

**Livro de Receitas** 1

**Álbum de Fotografias** 4

**Caderno de Músicas** 1

**Quero inserir uma nova informação!**

Figura 4.8: Formulário da Sessão Pessoal

Inclusão de Sessões

## CONSTRUTOR DE SITES PESSOAIS UNIVERSAIS

**SESSÕES** Pessoal Quero uma sessão que não está disponível!

Pessoal

**SESSÃO PESSOAL** X

**INSERINDO UMA NOVA INFORMAÇÃO NA MINHA SESSÃO PESSOAL**

Nome:  Nova

Conteúdo:  Gravar

Tipo:  Fechar

Nível de Acesso: 4 - Amigos e Família

Figura 4.9: Solicitação de nova informação na Sessão Pessoal

## 4.2 Sessão Profissional

As informações agrupadas estão demonstradas na Fig. 4.10, além de um *link* para uma *Home Page* Profissional que o indivíduo já tenha desenvolvido e **documentos profissionais**, como certificados, diplomas, etc, também podem ser disponibilizados, na forma eletrônica. Assim, quando houver concursos ou licitações o profissional não precisa tirar cópias de seus documentos profissionais, dirigir-se a um cartório para autenticá-los e arcar com os custos a cada vez. Com essa ferramenta, os documentos apresentados já possuem validade jurídica, pois passaram por um processo de verificação referente à autenticidade, integridade, confidencialidade e não-repúdio, validados por uma protocolizadora de documentos eletrônicos, onde data e hora foram anexados aos mesmos registrando sua existência e integridade a partir daquele momento. Então, mesmo se uma pessoa não tiver acesso aos seus documentos originais, poderá participar de tais eventos porque há um meio de disponibilizar cópias legalmente válidas através do *Site* Pessoal.

The image shows a software window titled "CONSTRUTOR DE SITES PESSOAIS UNIVERSAIS" with a sub-tab "SESSÃO PROFISSIONAL". The interface includes a legend for access levels (1-6) and a form with the following fields and options:

- Profissão:** Analista de Sistemas (dropdown)
- Titulação:** Mestre em fase de conclusão (dropdown)
- Empresa onde trabalha:** Autônoma (dropdown)
- Cargo ocupado:** Analista de Sistemas (dropdown)
- Idiomas:** Inglês (reconhecimento e leitura) (dropdown)
- Número da carteira de trabalho:** 2234564433 (dropdown)
- Número de registro profissional:** (dropdown)
- Atividades desenvolvidas:**
  - Análise de sistemas (dropdown)
  - Web design
  - Programação em Delphi
  - Banco de dados

Additional features include buttons for "Documentos Profissionais", "Currículo", "Home Page", and "Quero inserir uma nova informação!". The bottom of the window contains a row of buttons: "Alterar", "Salvar", "Visualizar", "Excluir", "Incluir", and "Fechar".

Figura 4.10: Formulário da Sessão Profissional

Nesta sessão, há um espaço para o **Currículo** que pode ser um *link* para um que já existe no *site* de alguma empresa especializada ou o modelo proposto no *site*, que tem

os seguinte grupos de informação: identificação, formação, experiência, prêmios, idiomas, orientações de trabalhos de pesquisa, participação em eventos, publicações, foto e, uma ferramenta inovadora, o vídeo digital. Ele é gravado na intenção de representar alguém no momento da entrevista e pode ser elaborado com base em parâmetros específicos disponibilizados pela empresa solicitadora.

Assim como nas outras sessões, também é possível a inclusão de novas informações.

### 4.3 Sessão Saúde

Nesta sessão o titular do *site* pode escolher entre apenas colocar um *link* para seu prontuário médico que já está pronto em um *site* especializado ou também preencher algumas informações sobre sua saúde, incluindo pessoas que devem ser encontradas em situações críticas, como pode ser observado na Fig. 4.11.

Exames laboratoriais e de imagem também podem ser armazenados nesta sessão.

Assim como nas sessões Pessoal e Profissional, descritas anteriormente, cada informação deve ter o nível de acesso indicado individualmente.

Figura 4.11: Formulário da Sessão Saúde.

## 4.4 Sessão de Demanda

Esta é a sessão, denominada QUERO, é destinada a colocar os produtos ou serviços que se quer comprar ou contratar. No exemplo da Fig. 4.12, é possível observar o preenchimento do formulário de busca de uma babá. Nota-se que não há necessidade de informar todos os campos disponíveis, mesmo porque alguns têm mais sentido quando colocados à venda, como as informações de contato, por exemplo.

Vale lembrar, que cada item tem seu próprio descritor, com as características específicas do mesmo. Portanto, cabe ao administrador do Portal Pessoal Universal a criação e disponibilização de descritores do maior número possível de produtos ou serviços. Sugestões ou pedidos também são aceitos, bastando o colaborador enviá-los por *e-mail* ao administrador do Portal.



The image shows a web browser window titled "Inclusão de Sessões" with a main header "CONSTRUTOR DE SITES PESSOAIS UNIVERSAIS". Below the header, there is a section for "SESSÕES" with a dropdown menu set to "Quero" and a button that says "Quero uma sessão que não está disponível!". Underneath, there are tabs for "Pessoal", "Profissional", "Saúde", and "Quero". The "SESSÃO QUERO" form is highlighted in yellow and contains the following fields and options:

- Produtos ou Serviços:
- Nome do Produto/Serviço:
- Nome da Bebê:
- Faixa etária:  de 15 a 20 anos  de 21 a 35 anos  de 36 a 50 anos
- Idade:
- Telefone:
- E-mail:
- Endereço: Rua:  Bairro:  Cidade:  CEP:  Complemento:
- Alcance ou Experiência com crianças:  até 1 ano  de 1 a 5 anos  de 5 a 10 anos  acima de 10 anos  qualquer idade
- Referências:
- Preço:
- Número de respostas:

At the bottom of the form, there are buttons for "Alterar", "Salvar", "Visualizar", "Excluir", "Incluir", and "Fechar".

Figura 4.12: Formulário da Sessão Quero, exemplificando o preenchimento do formulário para procurar por uma babá



## 4.5 Sessão de Oferta

Da mesma forma que a sessão Quero, aqui, na sessão TENHO, também são colocados os descritores dos objetos ou serviços, mas com o intuito de serem vendidos. O titular do *site* pode optar por esperar que o agente de compra de outra pessoa o encontre ou, se tiver pressa, decidir acionar seu próprio agente de venda que sai pela grande rede em busca de pessoas que queiram comprar seu produto.

A Fig. 4.13 demonstra o preenchimento do descritor de uma bicicleta que está sendo colocada à venda. Também se observa, que um descritor colocado na sessão Tenho tem mais características preenchidas. Isso é feito para que um potencial comprador tenha o maior número possível de detalhes sobre o produto em questão e assim ter melhores condições para tomar sua decisão.

The screenshot shows a web application window titled 'CONSTRUTOR DE SITES PESSOAIS UNIVERSAIS'. The main section is 'SESSÕES' with a dropdown menu set to 'Tenho' and a message 'Quero uma sessão que não está disponível!'. Below this are tabs for 'Pessoa', 'Profissional', 'Saúde', 'Quero', and 'Tenho'. The 'SESSÃO TENHO' form is active, with a 'Procurar' button. The form fields include:
 

- Produtos ou Serviços:
- Nome do Produto/Serviço:
- Cor:
- Nº de marchas:
- Fabricante:
- Tempo de uso:
- Altura:  Comprimento:  Largura:
- Estado de Mercado para Venda:
  - Não está à venda
  - Venda ou aceitar ofertas em circunstâncias especiais
  - O preço está fixado. O primeiro que pagar, leva
  - Faça uma oferta
  - Preciso vender imediatamente
  - Maior oferta até certa data leva
- Tipo:  Passeio  Comida  Esportística
- Estado atual:  Ótimo  Regular  Bom  Precisa de ajustes
- Condições de pagamento:
- Preço:
- Número de respostas:  Escolha a área de busca:

 At the bottom, there are buttons for 'Alterar', 'Salvar', 'Visualizar', 'Excluir', 'Incluir', and 'Fechar'.

Figura 4.13: Formulário da Sessão Tenho, demonstrando o preenchimento do descritor de uma bicicleta.

Os detalhes sobre o funcionamento das sessões estão descritos no capítulo 3, onde o Modelo Proposto é explanado.

## **5 CONSIDERAÇÕES FINAIS**

A idéia de desenvolver um modelo genérico para *Sites* Pessoais surgiu com o intuito de estimular as pessoas a utilizar a Internet como um meio de comunicação onde podem armazenar informações sobre sua vida pessoal e profissional de forma segura e com níveis de acesso. Através deste instrumento, pessoas ficam habilitadas a participar de determinados eventos que necessitem da apresentação de documentos específicos, os quais normalmente estão bem guardados em suas residências. Em casos críticos onde há risco de saúde, informações de grande valor podem ser encontradas no *Site* Pessoal. Neste mesmo espaço, é possível encontrar produtos ou serviços para comprar e vender de uma maneira simples, rápida e confortável, com o auxílio de agentes de *software*.

Nos dias de hoje percebe-se o aumento significativo de situações de violência, como assaltos, seqüestros e outras desse gênero, onde há o roubo de documentos pessoais. Esta é uma ocorrência que gera muita insegurança e o incômodo de ficar sem tais documentos. Mas, com essa nova ferramenta, até que seja atendida a solicitação de segunda via, alguém que esteja nesta situação pode disponibilizá-los no formato eletrônico, com a devida validade jurídica, se forem solicitados.

O modelo proposto neste documento fundamenta-se em uma estrutura genérica, onde as informações são divididas em sessões e disponibilizadas de acordo com níveis de acesso. Os *Sites* Pessoais são criados com o auxílio de um *software* Construtor de *Sites* Pessoais Universais (modelado neste documento), através da disponibilização de sessões e conteúdos predefinidos. A flexibilidade é um aspecto que se destaca, pois é possível acrescentar novas informações às sessões apresentadas, assim como criar novas sessões que sejam de interesse de alguém em particular. Desta forma, o *Site* Pessoal pode ajustar-se à realidade de qualquer indivíduo. É importante destacar que esta ferramenta apresenta-se como um meio para a publicação e interação da informação, atuando como propriedade intelectual, incentivando assim, a popularização da cultura digital.

A validação do modelo foi realizada através da elaboração de cinco sessões predefinidas – pessoal, profissional, saúde, de demanda e de oferta – onde podem ser organizadas diversas informações sobre uma pessoa. As três primeiras sessões agrupam informações sigilosas e verificou-se, durante o desenvolvimento desta dissertação, que a segurança é o aspecto fundamental para a viabilidade do *Site* Pessoal. No entanto, este requisito por si só apresenta-se como tema para um novo trabalho devido à complexidade envolvida na disponibilização e controle dos níveis de acesso. Por este motivo, num primeiro momento foi adotada a tecnologia de certificados digitais para permitir que somente pessoas autorizadas possam obter as informações contidas no *Site* Pessoal, bem como a protocolização de documentos, com o objetivo de comprovar a existência dos mesmos a partir de determinado dia e horário, ou seja, de garantir a integridade, veracidade e validade dos mesmos.

Já as sessões de demanda e de oferta, são destinadas a disponibilizar produtos ou serviços que se deseja comprar e vender, respectivamente. Para executar estas funções, foi identificada e sugerida a tecnologia de agentes de *software* para procurar tais itens e apresentar ao proprietário do *Site* Pessoal as informações de contato com potenciais compradores ou vendedores de seus produtos ou serviços. Devido ao nível de complexidade envolvido, a implantação desta tecnologia merece estudos mais criteriosos e aprofundados e, portanto, fica indicada para estudos futuros, pois se apresenta fora do escopo deste trabalho. Nestas sessões a segurança não se faz necessária, pois precisam ser de domínio público para que os itens sejam encontrados.

Outra indicação se faz necessária: trata-se da verificação da possibilidade de organizar, armazenar e resgatar as informações de um Banco de Dados através da linguagem XML. O fato do objeto de estudo desta dissertação ser completamente flexível, onde o usuário constrói o *Site* Pessoal de acordo com seus interesses, pode ter como resultado *Sites* Pessoais completamente diferentes quanto à variedade de novas sessões que podem ser criadas *on-line*. E para fazer a representação do conteúdo destas, os modelos relacional, orientado a objetos ou, até mesmo, o relacional-objeto (estudado neste documento) podem não ser a escolha mais adequada. É neste momento que a linguagem XML tem sua indicação máxima, pois tem uma abordagem específica para representar dados semi-estruturados (aqueles que têm uma representação estrutural irregular), nos quais a estrutura é conhecida no instante de sua criação (SACCOL,

2002). Brian C. Reed (REED, 2002), escreve em seu artigo sobre duas abordagens possíveis para a parceria XML e BD. A primeira refere-se ao mapeamento dos dados de um documento XML para tipos específicos que são organizados em estrutura tabulares, bem como a definição de funções e procedimentos para manipular esses novos tipos de dados. A segunda abordagem trabalha com os documentos XML em sua forma nativa, realizando as operações de escrita, leitura e manutenção sobre os dados sem haver qualquer tipo de transferência ou mapeamento para outra estrutura. Portanto, fica registrada aqui a sugestão da aplicação da linguagem XML para a implementação do Portal Pessoal Universal.

O desenvolvimento deste trabalho proporcionou a oportunidade de expandir o conhecimento em áreas diferentes, como internet, organização e armazenamento de informações pessoais, segurança e agentes de *software* reunindo-as num único projeto. Espera-se que com a disponibilização de um serviço como o proposto aqui, mais pessoas sintam-se à vontade para participar da Internet, onde aplicações de diversas naturezas são oferecidas e utilizadas com o intuito de facilitar a vida das pessoas que fazem uso delas.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

### 6.1 Bibliografias

(BOOCH, 2002) Quality Software and the Unified Modeling Language. By Grady Booch. Disponível em: <http://www.rational.com/products/whitepapers/285.jsp>. Acesso em: 07/05/2002.

(BORTOLI, 2002) BORTOLI, DeJane Luiza. O Documento Eletrônico no Ofício de Registro Civil de Pessoas Naturais. 2002. 113f. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.inf.ufsc.br/~custodio/orientacao.html>. Acesso em: 10/07/2002.

(Ciência e Tecnologia para a Construção..., 1999) Ciência e Tecnologia para a Construção da Sociedade da Informação. Ministério de Ciência e Tecnologia – Programa da Sociedade da Informação. Editores: Carlos José Pereira de Lucena, Ivan Moura Campos, Silvio Lemos Meira. 1999. Disponível em: [http://www.mct.gov.br/Temas/Socinfo/CeT\\_socinfo.pdf](http://www.mct.gov.br/Temas/Socinfo/CeT_socinfo.pdf). Acesso em: 09/03/2001.

(Conceptual, Logical..., 2002) Conceptual, Logical and Physical design of Persistent Data using UML. Disponível em: <http://www.rational.com/products/whitepapers/411.jsp>. Acesso em: 07/05/2002.

(COSER, 1999) COSER, Adriano. Utilização de Agentes Inteligentes no Trabalho Colaborativo via Internet. 1999. 134f. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

(FARACO, 1998) FARACO, Rafael A. – Uma Arquitetura de Agentes para Negociação dentro do Domínio do Comércio Eletrônico. 1998. 100f. Cap. 2. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis. Disponível em:

<http://www.eps.ufsc.br/disserta98/faraco>. Acessado em 16/07/2002

(FERREIRA, 1999) FERREIRA, Aurélio Buarque de Holanda. Novo Aurélio Século XXI. Rio de Janeiro: Nova Fronteira, 1999.

(FLEISCHHAUER, 1996) FLEISCHHAUER, Luciana I. A. – O Uso da Tecnologia de Agentes na Integração da Programação de Produção – 1996. Cap. 3. Disponível em:

<http://www.inf.ufsc.br/~gauthier/SIMLages/SIMLages.html>. Acesso em 16/07/2002.

(FOWLER, SCOTT, 2000) FOWLER, Martin; SCOTT, Kendall. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. Porto Alegre: Bookman, 2000.

(IGNACZAK, 2002) IGNACZAK, Luciano. Um novo modelo de Infra-estrutura de Chaves Públicas para uso no Brasil utilizando aplicativos com o código fonte aberto.

2002. 126f. Dissertação (Mestrado em Ciências da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.inf.ufsc.br/~custodio/orientacao.html>. Acesso em: 10/07/2002.

(LARMAN, 2000) LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos. Porto Alegre: Bookman, 2000.

(LEMAY, 1998) LEMAY, Laura; PERKINS. Aprenda em 21 Dias Java. Rio de Janeiro: Campus, 1998.

(Livro Verde da Sociedade da Informação..., 2000) Livro Verde da Sociedade da Informação no Brasil. Ministério de Ciência e Tecnologia – Programa da Sociedade da Informação. 2000. Disponível em:

<http://www.mct.gov.br/Temas/Socinfo/livroverde.htm>. Acesso em: 04/08/2000.

(LOWERY, 2001) LOWERY, Joseph. Dreaweaver 4: a bíblia. Rio de Janeiro: Campus, 2001.

(NOTOYA, 2002) NOTOYA, Adriana Elissa. IARSDE – Infra-estrutura de Armazenamento e Recuperação Segura de Documentos Eletrônicos: validade do documento eletrônico por tempo indeterminado. 2002. 119f. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.inf.ufsc.br/~custodio/orientacao.html>. Acesso em: 10/07/2002.

(OLIVEIRA, 2000) OLIVEIRA, Jayr Figueiredo de. Sistemas de Informação: um enfoque gerencial inserido no contexto empresarial e tecnológico. São Paulo: Érica, 2000.

(PASQUAL, 2001) PASQUAL, Everton Schonardie. IDDE – Uma Infra-estrutura para a Datação de Documentos Eletrônicos. 2001. 102f. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.inf.ufsc.br/~custodio/orientacao.html>. Acesso em: 10/07/2002.

(Revista MEDICINA Conselho Federal, 2001) Debate: Prontuário Eletrônico. Revista MEDICINA Conselho Federal. Ano XVI, n° 125, 5p, Janeiro, 2001.

(ROCHA, 2001) ROCHA, João Luiz Francalacci. Proteção de Software por Certificação Digital. 2001. 109f. Dissertação (Mestrado em Ciências da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.inf.ufsc.br/~custodio/orientacao.html>. Acesso em: 10/07/2002.

(SACCOL, 2001) SACCOL, Deise de Brum. Materialização de Visões XML. 2001. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal do Rio Grande do Sul. Disponível em: <http://www.bilbiloteca.ufrgs.br/bibliotecadigital/2002-1/tese-inf-0326029.pdf>. Acesso em 21/10/2002.

(SAVIO, 2000) SAVIO, Luciano José. Um Modelo Basado em Agentes para Suporte a Automação de Serviços. 2000. 110f. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

(SIEGEL, 2000) SIEGEL, David. Futurize sua Empresa. São Paulo: Futura, 2000.

(SILBERSCHATZ, KORTH, SUDARSHAN, 1999) SILBERSCHATZ, Abraham; KORTH, Henry F. e SUDARSHAN, S. Sistema de Banco de Dados. São Paulo: Makron Books, 1999. 3. ed.

(SOUZA, 1996) SOUZA, Eliane Moreira Sá de. Uma Estrutura de Agentes para Assessoria na Internet. 1996. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.eps.ufsc.br/disserta96/eliane/index/index.htm>. Acesso em 16/07/2002.

(TORRES, COZER, 2000) TORRES, Gabriel, COZER, Alberto. Alavancando Negócios na Internet. Rio de Janeiro: Axcel Books, 2000.

(REED, 2002) REED, Brian C. Can XML Web Services Offer a Standard Across Databases? Mundo OO. **DevX**. [http://archive.devx.com/xml/articles/dd\\_dbstandard/dbstandard-1.asp](http://archive.devx.com/xml/articles/dd_dbstandard/dbstandard-1.asp). Acesso em: 21/10/2002.

## **6.2 Sites Consultados**

Aol. <http://www.aol.com.br>

Bol. <http://www.bol.com.br>



Bry Tecnologia - Protocolizadora Digital de Documentos Eletrônicos. Disponível em:  
[http://www.bry.com.br/folders/bry\\_pdde Rack\\_pt.pdf](http://www.bry.com.br/folders/bry_pdde Rack_pt.pdf). Acesso em: 11/07/2002.

Bry Tecnologia – Kit de Desenvolvimento Para Protocolização Digital. Disponível em:  
[http://www.bry.com.br/folders/bry\\_pdde\\_sdk\\_pt.pdf](http://www.bry.com.br/folders/bry_pdde_sdk_pt.pdf). Acesso em: 11/07/2002.

Bry Tecnologia – BRy Signer. Disponível em:  
[http://www.bry.com.br/folders/bry\\_signer.pdf](http://www.bry.com.br/folders/bry_signer.pdf). Acesso em: 11/07/2002.

Bry Tecnologia – Kit de Desenvolvimento Para Assinatura Digital. Disponível em:  
[http://www.bry.com.br/folders/bry\\_signer\\_sdk\\_pt.pdf](http://www.bry.com.br/folders/bry_signer_sdk_pt.pdf) . Acesso em: 11/07/2002.

Bry Tecnologia – Segurança na Internet e Certificação Digital. Disponível em:  
[http://www.bry.com.br/download/BRy\\_Artigo\\_Criptografia.pdf](http://www.bry.com.br/download/BRy_Artigo_Criptografia.pdf). Acesso em:  
11/07/2002.

Dicionário UNIVERSAL da Língua Portuguesa *on-line*. Disponível em:  
<http://www.priberam.pet/DLPO/> Acesso em: 21/10/2002.

CertSign - Certificação Digital. Disponível em:  
[http://www.certisign.com.br/help\\_email/index.html](http://www.certisign.com.br/help_email/index.html). Acesso em: 14/04/2001.

Home Page Grátis - hpG. <http://www.hpg.com.br>

Medida Provisória 2200-2. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/MPV/2200-2.htm](http://www.planalto.gov.br/ccivil_03/MPV/2200-2.htm) Acesso em: 10/07/2002

# **ANEXO 1 - LISTA DE CATEGORIA DE CONCEITOS**

As seguintes sentenças, propostas por Larman [Lar00], devem ser aplicadas a cada caso de uso com a finalidade de identificar conceitos relevantes que farão parte do modelo conceitual de um sistema:

1. Objetos físicos ou tangíveis;
2. Especificações de projetos ou descrições de coisas;
3. Lugares;
4. Transações;
5. Linhas de itens de transações;
6. Papéis desempenhados por pessoas;
7. Contêiners de outras coisas;
8. Coisas em um contêiner;
9. Outros sistemas de computador ou dispositivos eletromecânicos externos ao sistema abordado;
10. Conceitos de substantivos abstratos;
11. Organizações;
12. Eventos;
13. Processo (freqüentemente não-representados como conceitos, embora possam ser representados como tal);
14. Regras e estratégias;
15. Catálogos
16. Registros financeiros, trabalhistas, de contratos, de assuntos legais;
17. Serviços e instrumentos financeiros;
18. Manuais e livros.

# ANEXO 2 – MEDIDA PROVISÓRIA 2200-2



## **Presidência da República Casa Civil Subchefia para Assuntos Jurídicos**

### MEDIDA PROVISÓRIA Nº 2.200-2, DE 24 DE AGOSTO DE 2001.

#### **Em tramitação**

Institui a Infra-Estrutura de Chaves Públicas Brasileira - ICP-Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências.

**O PRESIDENTE DA REPÚBLICA**, no uso da atribuição que lhe confere o art. 62 da Constituição, adota a seguinte Medida Provisória, com força de lei:

Art. 1º Fica instituída a Infra-Estrutura de Chaves Públicas Brasileira - ICP-Brasil, para garantir a autenticidade, a integridade e a validade jurídica de documentos em forma eletrônica, das aplicações de suporte e das aplicações habilitadas que utilizem certificados digitais, bem como a realização de transações eletrônicas seguras.

Art. 2º A ICP-Brasil, cuja organização será definida em regulamento, será composta por uma autoridade gestora de políticas e pela cadeia de autoridades certificadoras composta pela Autoridade Certificadora Raiz - AC Raiz, pelas Autoridades Certificadoras - AC e pelas Autoridades de Registro - AR.

Art. 3º A função de autoridade gestora de políticas será exercida pelo Comitê Gestor da ICP-Brasil, vinculado à Casa Civil da Presidência da República e composto por cinco representantes da sociedade civil, integrantes de setores interessados, designados pelo Presidente da República, e um representante de cada um dos seguintes órgãos, indicados por seus titulares:

I - Ministério da Justiça;

II - Ministério da Fazenda;

III - Ministério do Desenvolvimento, Indústria e Comércio Exterior;

IV - Ministério do Planejamento, Orçamento e Gestão;

V - Ministério da Ciência e Tecnologia;

VI - Casa Civil da Presidência da República; e

VII - Gabinete de Segurança Institucional da Presidência da República.

§ 1º A coordenação do Comitê Gestor da ICP-Brasil será exercida pelo representante da Casa Civil da Presidência da República.

§ 2º Os representantes da sociedade civil serão designados para períodos de dois anos, permitida a recondução.

§ 3º A participação no Comitê Gestor da ICP-Brasil é de relevante interesse público e não será remunerada.

§ 4º O Comitê Gestor da ICP-Brasil terá uma Secretaria-Executiva, na forma do regulamento.

Art. 4º Compete ao Comitê Gestor da ICP-Brasil:

I - adotar as medidas necessárias e coordenar a implantação e o funcionamento da ICP-Brasil;

II - estabelecer a política, os critérios e as normas técnicas para o credenciamento das AC, das AR e dos demais prestadores de serviço de suporte à ICP-Brasil, em todos os níveis da cadeia de certificação;

III - estabelecer a política de certificação e as regras operacionais da AC Raiz;

IV - homologar, auditar e fiscalizar a AC Raiz e os seus prestadores de serviço;

V - estabelecer diretrizes e normas técnicas para a formulação de políticas de certificados e regras operacionais das AC e das AR e definir níveis da cadeia de certificação;

VI - aprovar políticas de certificados, práticas de certificação e regras operacionais, credenciar e autorizar o funcionamento das AC e das AR, bem como autorizar a AC Raiz a emitir o correspondente certificado;

VII - identificar e avaliar as políticas de ICP externas, negociar e aprovar acordos de certificação bilateral, de certificação cruzada, regras de interoperabilidade e outras formas de cooperação internacional, certificar, quando for o caso, sua compatibilidade com a ICP-Brasil, observado o disposto em tratados, acordos ou atos internacionais; e

VIII - atualizar, ajustar e revisar os procedimentos e as práticas estabelecidas para a ICP-Brasil, garantir sua compatibilidade e promover a atualização tecnológica do sistema e a sua conformidade com as políticas de segurança.

Parágrafo único. O Comitê Gestor poderá delegar atribuições à AC Raiz.

Art. 5º À AC Raiz, primeira autoridade da cadeia de certificação, executora das Políticas de Certificados e normas técnicas e operacionais

aprovadas pelo Comitê Gestor da ICP-Brasil, compete emitir, expedir, distribuir, revogar e gerenciar os certificados das AC de nível imediatamente subsequente ao seu, gerenciar a lista de certificados emitidos, revogados e vencidos, e executar atividades de fiscalização e auditoria das AC e das AR e dos prestadores de serviço habilitados na ICP, em conformidade com as diretrizes e normas técnicas estabelecidas pelo Comitê Gestor da ICP-Brasil, e exercer outras atribuições que lhe forem cometidas pela autoridade gestora de políticas.

Parágrafo único. É vedado à AC Raiz emitir certificados para o usuário final.

Art. 6º Às AC, entidades credenciadas a emitir certificados digitais vinculando pares de chaves criptográficas ao respectivo titular, compete emitir, expedir, distribuir, revogar e gerenciar os certificados, bem como colocar à disposição dos usuários listas de certificados revogados e outras informações pertinentes e manter registro de suas operações.

Parágrafo único. O par de chaves criptográficas será gerado sempre pelo próprio titular e sua chave privada de assinatura será de seu exclusivo controle, uso e conhecimento.

Art. 7º Às AR, entidades operacionalmente vinculadas a determinada AC, compete identificar e cadastrar usuários na presença destes, encaminhar solicitações de certificados às AC e manter registros de suas operações.

Art. 8º Observados os critérios a serem estabelecidos pelo Comitê Gestor da ICP-Brasil, poderão ser credenciados como AC e AR os órgãos e as entidades públicos e as pessoas jurídicas de direito privado.

Art. 9º É vedado a qualquer AC certificar nível diverso do imediatamente subsequente ao seu, exceto nos casos de acordos de certificação lateral ou cruzada, previamente aprovados pelo Comitê Gestor da ICP-Brasil.

Art. 10. Consideram-se documentos públicos ou particulares, para todos os fins legais, os documentos eletrônicos de que trata esta Medida Provisória.

§ 1º As declarações constantes dos documentos em forma eletrônica produzidos com a utilização de processo de certificação disponibilizado pela ICP-Brasil presumem-se verdadeiros em relação aos signatários, na forma do [art. 131 da Lei nº 3.071, de 1º de janeiro de 1916](#) - Código Civil.

§ 2º O disposto nesta Medida Provisória não obsta a utilização de outro meio de comprovação da autoria e integridade de documentos em forma eletrônica, inclusive os que utilizem certificados não emitidos pela ICP-Brasil, desde que admitido pelas partes como válido ou aceito pela pessoa a quem for oposto o documento.

Art. 11. A utilização de documento eletrônico para fins tributários atenderá, ainda, ao disposto no [art. 100 da Lei nº 5.172, de 25 de outubro de 1966](#) - Código Tributário Nacional.

Art. 12. Fica transformado em autarquia federal, vinculada ao Ministério da Ciência e Tecnologia, o Instituto Nacional de Tecnologia da Informação - ITI, com sede e foro no Distrito Federal.

Art. 13. O ITI é a Autoridade Certificadora Raiz da Infra-Estrutura de Chaves Públicas Brasileira.

Art. 14. No exercício de suas atribuições, o ITI desempenhará atividade de fiscalização, podendo ainda aplicar sanções e penalidades, na forma da lei.

Art. 15. Integrarão a estrutura básica do ITI uma Presidência, uma Diretoria de Tecnologia da Informação, uma Diretoria de Infra-Estrutura de Chaves Públicas e uma Procuradoria-Geral.

Parágrafo único. A Diretoria de Tecnologia da Informação poderá ser estabelecida na cidade de Campinas, no Estado de São Paulo.

Art. 16. Para a consecução dos seus objetivos, o ITI poderá, na forma da lei, contratar serviços de terceiros.

§ 1º O Diretor-Presidente do ITI poderá requisitar, para ter exercício exclusivo na Diretoria de Infra-Estrutura de Chaves Públicas, por período não superior a um ano, servidores, civis ou militares, e empregados de órgãos e entidades integrantes da Administração Pública Federal direta ou indireta, quaisquer que sejam as funções a serem exercidas.

§ 2º Aos requisitados nos termos deste artigo serão assegurados todos os direitos e vantagens a que façam jus no órgão ou na entidade de origem, considerando-se o período de requisição para todos os efeitos da vida funcional, como efetivo exercício no cargo, posto, graduação ou emprego que ocupe no órgão ou na entidade de origem.

Art. 17. Fica o Poder Executivo autorizado a transferir para o ITI:

I - os acervos técnico e patrimonial, as obrigações e os direitos do Instituto Nacional de Tecnologia da Informação do Ministério da Ciência e Tecnologia;

II - remanejar, transpor, transferir, ou utilizar, as dotações orçamentárias aprovadas na Lei Orçamentária de 2001, consignadas ao Ministério da Ciência e Tecnologia, referentes às atribuições do órgão ora transformado, mantida a mesma classificação orçamentária, expressa por categoria de programação em seu menor nível, observado o disposto no § 2º do [art. 3º da Lei nº 9.995, de 25 de julho de 2000](#), assim como o respectivo detalhamento por esfera orçamentária, grupos de despesa, fontes de recursos, modalidades de aplicação e identificadores de uso.

Art. 18. Enquanto não for implantada a sua Procuradoria Geral, o ITI será representado em juízo pela Advocacia Geral da União.

Art. 19. Ficam convalidados os atos praticados com base na Medida Provisória nº 2.200-1, de 27 de julho de 2001.

Art. 20. Esta Medida Provisória entra em vigor na data de sua publicação.

Brasília, 24 de agosto de 2001; 180º da Independência e 113º da República.

FERNANDO HENRIQUE CARDOSO

*José Gregori*

*Martus Tavares*

*Ronaldo Mota Sardenberg  
Pedro Parente*

*Este texto não substitui o publicado no D.O.U. de 27.8.2001*

## ANEXO 3 - GLOSSÁRIO

**AC:** Autoridade Certificadora.

**AR:** Autoridade de Registro.

**Aplicação:** *software* ou sistema de computador.

**Browser:** *software* que permite navegar pela Internet.

**Cifragem:** codificação ou embaralhamento de uma informação.

**Chave:** senha.

**CSPU:** Construtor de *Sites* Pessoais Universais

**Decifragem:** o contrário de cifragem.

**Home Page:** página inicial de um *site*.

**Link:** conexão entre duas páginas da Internet.

**Protocolizar:** protocolar. Registrar ou escrever no protocolo.

**Protocolo:** cartão ou papeleta em que se anotam a data e o número de ordem com que um requerimento foi registrado no livro de protocolo e que serve como recibo.