

Instituto de Pesquisas Tecnológicas do Estado de São Paulo

ADALBERTO JOSÉ NERY

Projeto de um gerenciador de serviços aplicado ao domínio dos  
servidores de aplicação.

São Paulo

2005

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

ADALBERTO JOSÉ NERY

Projeto de um gerenciador de serviços aplicado ao domínio dos servidores de aplicação.

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, para obtenção do título de Mestre em Engenharia de Computação.

Área de concentração: Engenharia de Software

Orientadora: Dra. Edit Grassiani Lino de Campos.

São Paulo

Dezembro 2005

Ficha Catalográfica  
Elaborada pelo Centro de Informação Tecnológica do  
Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT

**N456p Nery, Adalberto José**

Projeto de um gerenciador de serviços aplicado ao domínio dos servidores de aplicação. / Adalberto José Nery. São Paulo, 2005.  
115p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Engenharia de Software.

Orientador: Prof. Dra. Edit Grassiani Lino de Campos

1. Gerenciador de serviços 2. Servidor de aplicação 3. Tese I. Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Centro de Aperfeiçoamento Tecnológico II. Título

## RESUMO

Atualmente existem indícios fortes de que, em um futuro bem próximo, a maioria dos processos de negócios informatizados será reutilizada de forma interna às empresas e se expandirá de forma global via Internet. Em virtude do aumento de consumo desses processos que podem ser considerados serviços eletrônicos e de um cenário econômico global de contenção de despesas que atinge as empresas, uma das possíveis tomadas de ações, eleita por este trabalho, é melhorar a gestão desses recursos providos pelos servidores de aplicação de maneira a superar algumas deficiências. Entre elas estão: a insuficiência das informações sobre os serviços e a falta de mecanismos para divulgá-las e pesquisá-las; a ausência de registro e controle dos componentes utilizados pelos serviços, das configurações e versões dos serviços e dos problemas que freqüentemente ocorrem com os mesmos (falhas, travamentos, alta utilização de CPU, entre outros), além das respectivas soluções; e a falta de registro, recuperação e divulgação das modificações que ocorreram com os serviços, durante sua vida útil. Um gerenciador de serviços tem o intuito de superar estas carências, melhorando o controle e a gestão. Com este intuito, este trabalho apresenta uma ferramenta que gerencia os diversos serviços providos por um servidor de aplicação, além de prover um *site* para pesquisas do acervo e de dispor sua base de conhecimento para a geração de relatórios ou consultas operacionais e gerenciais.

Palavras-chave: Servidor de aplicação; Serviços; Administração e gerenciamento de serviços; Gerência de serviços; Gestão de configuração e versões; Processos de negócio; Gerenciador de serviços.

## ABSTRACT

Project of a service manager applied to the application servers' domain.

There are strong indications that, in the near future, the majority of the computerized business-oriented processes will be reused within the businesses, and will expand globally throughout the Internet. Due to the increase of consumption of these processes considered electronic services and influenced by the global economic expense reduction, one of the possible actions, elected in this work, is to improve the management of resources provided by application servers in order to overcome some deficiencies. Some of these deficiencies are: insufficient information about the services and lack of publishing and search mechanisms; lack of registry and control over components used by the services, service configuration and versions or problem raised (failures, blockings, high CPU utilization) and their solutions, and service update history. This work aims at overcoming these points by presenting a management tool for services hosted by an application server. The tool provides access and control of a service knowledge base containing and renders operational and managerial reports, as well as query resolution.

Keywords: Application Server; Services; Service Administration and Management; Service Management; Configuration Management and Version Control; Business Processes; Service Manager.

## Lista de ilustrações

|             |  |    |
|-------------|--|----|
| Figura 1.1  | Arquitetura cliente – servidor de três camadas.  | 1  |
| Figura 2.1  | Empresas fornecedoras de Servidores de Aplicação.  | 9  |
| Figura 2.2  | Arquitetura distribuída em N camadas   | 11 |
| Figura 2.3  | Arquitetura dos sistemas J2EE.   | 12 |
| Figura 3.1  | Modelo de três camadas.  | 22 |
| Figura 3.2  | Integração do Gerenciador de Serviços com o Servidor de Aplicações.  | 23 |
| Figura 3.3  | Visão conceitual dos elementos que participam do assunto serviços.   | 24 |
| Figura 3.4  | Diagrama de Classes do Gerenciador de serviços.  | 25 |
| Figura 3.5  | Diagrama de Classes da API de interface entre Gerenciador de serviços e o Servidor de aplicação            | 26 |
| Figura 4.1  | Diagrama de Distribuição e Componentes do Sistema.   | 29 |
| Figura 4.2  | Diagrama Entidade – Relacionamento do Gerenciador de serviços.   | 30 |
| Figura 4.3  | Objetos visuais que compõem as telas   | 32 |
| Figura 4.4  | Direção e sentido da navegação entre os objetos da tela.   | 32 |
| Figura 4.5  | Forma de apresentação dos campos texto e das caixas de texto.  | 33 |
| Figura 4.6  | Elementos gráficos que compõem a parte inicial da tela.  | 33 |
| Figura 4.7  | Elementos gráficos que compõem a parte final da tela.  | 33 |
| Figura 4.8  | Serviço de autenticação.   | 36 |
| Figura A.1  | Diagrama de caso de uso - Cadastramento e manutenção do acervo de serviços.                                | 59 |
| Figura A.2  | Diagrama de caso de uso - Cadastramento de serviços adquiridos de terceiros.                               | 60 |
| Figura A.3  | Diagrama de caso de uso - Cadastramento e manutenção da interface do serviço.                              | 61 |
| Figura A.4  | Diagrama de caso de uso - Cadastramento e manutenção dos componentes utilizados pelos serviços.            | 63 |
| Figura A.5  | Diagrama de caso de uso - Cadastramento e manutenção dos problemas e soluções que ocorrem com os serviços. | 64 |
| Figura A.6  | Diagrama de caso de uso - Estorno das modificações que ocorreram com os serviços.                          | 65 |
| Figura A.7  | Diagrama de caso de uso - Alteração do Estado do serviço.  | 66 |
| Figura A.8  | Diagrama de caso de uso - Cadastramento dos usuários responsáveis.   | 68 |
| Figura A.9  | Diagrama de caso de uso - Administração da data de expiração dos serviços.                                 | 69 |
| Figura A.10 | Diagrama de caso de uso - Administração dos problemas não solucionados relativos aos serviços.             | 69 |
| Figura A.11 | Diagrama de caso de uso - Administração do estado dos serviços.  | 70 |
| Figura A.12 | Diagrama de caso de uso - Divulgação automática de informações sobre os serviços.                          | 71 |
| Figura A.13 | Diagrama de caso de uso - Análise das informações através de ferramentas externas.                         | 72 |
| Figura D.1  | Diagrama de Sequência – Cadastramento de Serviços.   | 80 |
| Figura D.2  | Diagrama de Sequência – Cadastramento de Serviços – Exclusão.  | 81 |
| Figura D.3  | Diagrama de Sequência – Cadastramento de Serviços de terceiros.  | 81 |
| Figura D.4  | Diagrama de Sequência – Cadastramento da interface do serviço.   | 82 |

|             |  |     |
|-------------|--|-----|
| Figura D.5  | Diagrama de Seqüência – Cadastramento da interface do serviço – Exclusão.                            | 82  |
| Figura D.6  | Diagrama de Seqüência – Cadastramento dos componentes utilizados pelos serviços.                     | 83  |
| Figura D.7  | Diagrama de Seqüência – Cadastramento dos componentes utilizados pelos serviços – Exclusão.          | 83  |
| Figura D.8  | Diagrama de Seqüência – Cadastramento dos problemas com os serviços.                                 | 84  |
| Figura D.9  | Diagrama de Seqüência – Cadastramento dos problemas com os serviços – Exclusão.                      | 84  |
| Figura D.10 | Diagrama de Seqüência – Estorno das modificações.  | 85  |
| Figura D.11 | Diagrama de Seqüência – Alteração do estado do serviço.  | 85  |
| Figura D.12 | Diagrama de Seqüência – Cadastramento de Responsáveis  | 86  |
| Figura D.13 | Diagrama de Seqüência – Cadastramento de Responsáveis –Exclusão.                                     | 86  |
| Figura D.14 | Diagrama de Seqüência –Administração da data de expiração dos serviços.                              | 87  |
| Figura D.15 | Diagrama de Seqüência – Administração dos problemas não solucionados relativos aos serviços.         | 87  |
| Figura D.16 | Diagrama de Seqüência – Administração do estado dos serviços.  | 88  |
| Figura D.17 | Diagrama de Seqüência – Analisa as informações contidas na base de dados através de um visualizador. | 88  |
| Figura E.1  | Protótipo de tela – Tela principal.  | 89  |
| Figura E.2  | Protótipo de tela – Cadastro de Serviços.  | 91  |
| Figura E.3  | Protótipo de tela – Cadastro de Interface de um serviço.   | 94  |
| Figura E.4  | Protótipo de tela - Cadastro dos componentes utilizados pelos serviços.                              | 95  |
| Figura E.5  | Protótipo de tela – Cadastro de Problemas.   | 97  |
| Figura E.6  | Protótipo de tela – Estorno das últimas modificações.  | 99  |
| Figura E.7  | Protótipo de tela – Altera Estado do Serviço.  | 101 |
| Figura E.8  | Protótipo de tela – Cadastro dos Responsáveis.   | 102 |



## Lista de tabelas

|             |  |     |
|-------------|--|-----|
| Tabela 3.1  | Lista das Classes que participam do projeto.   | 26  |
| Tabela 4.1  | Lista dos principais métodos.  | 35  |
| Tabela E.1  | Campos da Tela principal.  | 90  |
| Tabela E.2  | Comandos da Tela do Cadastro de Serviços.  | 91  |
| Tabela E.3  | Campos da Tela do Cadastro de serviços.  | 92  |
| Tabela E.4  | Comandos da Tela do Cadastro da interface do serviço.  | 94  |
| Tabela E.5  | Campos da Tela do Cadastro da interface do serviço.  | 94  |
| Tabela E.6  | Comandos da Tela do Cadastro dos componentes utilizados pelos serviços.                      | 96  |
| Tabela E.7  | Campos da Tela do Cadastro dos componentes utilizados pelos serviços.                        | 96  |
| Tabela E.8  | Comandos da Tela do Cadastro de Problemas.   | 97  |
| Tabela E.9  | Campos da Tela do Cadastro de Problemas.   | 98  |
| Tabela E.10 | Comandos da Tela do Estorno das últimas modificações.  | 99  |
| Tabela E.11 | Campos da Tela do Estorno das últimas modificações.  | 99  |
| Tabela E.12 | Comandos da Tela - Altera Estado do Serviço.   | 101 |
| Tabela E.13 | Campos da Tela - Altera Estado do Serviço.   | 101 |
| Tabela E.14 | Comandos da Tela do Cadastro dos Responsáveis.   | 103 |
| Tabela E.15 | Campos da Tela do Cadastro dos Responsáveis.   | 103 |
| Tabela F.1  | Contrato da classe Serviço, operação: registrar dados do serviço().                          | 104 |
| Tabela F.2  | Contrato da classe Serviço, operação: verificar serviços vencidos().                         | 104 |
| Tabela F.3  | Contrato da classe Serviço, operação: verificar serviços a vencer().                         | 105 |
| Tabela F.4  | Contrato da classe Interface, operação: registrar dados da interface().                      | 105 |
| Tabela F.5  | Contrato da classe Serviço, operação: registrar dados do serviço().                          | 106 |
| Tabela F.6  | Contrato da classe Modificação, operação: registrar modificação().                           | 106 |
| Tabela F.7  | Contrato da classe Problema, operação: registrar dados do problema().                        | 107 |
| Tabela F.8  | Contrato da classe Problema, operação: verificar problemas não concluídos().                 | 107 |
| Tabela F.9  | Contrato da classe Responsável, operação: registrar dados do responsável().                  | 108 |
| Tabela F.10 | Contrato da classe Responsável, operação: avisar responsável().                              | 108 |
| Tabela F.11 | Contrato da classe Servidor de aplicação, operação: atualizar dados no servidor aplicação(). | 109 |

## Lista de abreviaturas e siglas

|       |   |
|-------|---|
| A2A   | <i>Application to application.</i>                        |
| ACID  | Atomicidade; Consistência; Isolamento e Durabilidade.     |
| API   | <i>Application programming interface.</i>                 |
| AS    | <i>Application Servers.</i>                               |
| B2B   | <i>Business to business.</i>                              |
| BD    | Banco de dados.   |
| C2B   | <i>Client to business.</i>                                |
| COM   | <i>Microsoft Component Object Model.</i>                  |
| DB    | Database – Banco de Dados.                                |
| DSS   | <i>Decision Support System.</i>                           |
| EJB   | <i>Enterprise JavaBeans.</i>                              |
| GUI   | <i>Graphic User Interface.</i>                            |
| IHC   | Interface Homem-Computador.                               |
| IHM   | Interface Homem-Máquina.                                  |
| J2EE  | <i>Java 2 Enterprise Edition.</i>                         |
| JMX   | <i>Java Management Extension.</i>                         |
| JNDI  | <i>Java Naming and Directory Interface.</i>               |
| MSAP  | <i>Microsoft Application Platform.</i>                    |
| ODBC  | <i>Open DataBase Connectivity</i>                         |
| QoS   | <i>Quality of Service.</i>                                |
| OLTP  | <i>On-Line Transaction Processing</i>                     |
| SGBDR | Sistema Gerenciador de Banco de Dados Relacional.         |
| SMTP  | <i>Simple Mail Transport Protocol.</i>                    |
| UDDI  | <i>Universal Description, Discovery, and Integration.</i> |
| UML   | <i>Unified Modeling Language.</i>                         |
| W3C   | <i>World Wide Web Consortium.</i>                         |
| WAP   | <i>Wireless Application Protocol.</i>                     |
| WSDL  | <i>Web Services Description Language.</i>                 |
| XML   | <i>extensible Markup Language.</i>                        |

## Sumário

|  |           |
|--|-----------|
| <b>CAPÍTULO 1 .....</b>                                      | <b>1</b>  |
| 1 INTRODUÇÃO .....   | 1         |
| 1.1 MOTIVAÇÃO.....   | 1         |
| 1.2 OBJETIVO.....  | 3         |
| 1.3 RESULTADOS ESPERADOS E CONTRIBUIÇÕES.....                | 4         |
| 1.4 METODOLOGIA.....   | 4         |
| 1.5 ORGANIZAÇÃO DO TRABALHO .....                            | 5         |
| <b>CAPÍTULO 2 .....</b>                                      | <b>7</b>  |
| 2 CONCEITOS PRINCIPAIS E ESTADO DA ARTE .....                | 7         |
| 2.1 INTRODUÇÃO.....  | 7         |
| 2.2 SERVIÇOS.....  | 7         |
| 2.3 SERVIDOR DE APLICAÇÃO .....                              | 9         |
| 2.4 GERENCIAMENTO DE SERVIÇOS .....                          | 14        |
| 2.5 CONCLUSÃO.....   | 15        |
| <b>CAPÍTULO 3 .....</b>                                      | <b>17</b> |
| 3 LEVANTAMENTO, ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS .....  | 17        |
| 3.1 INTRODUÇÃO.....  | 17        |
| 3.2 REQUISITOS FUNCIONAIS. ....                              | 17        |
| 3.3 REQUISITOS NÃO FUNCIONAIS .....                          | 19        |
| 3.4 CENÁRIO DE USO .....                                     | 20        |
| 3.4.1 ATORES QUE PARTICIPAM DOS CASOS DE USO .....           | 20        |
| 3.4.2 TIPOS DE INTERAÇÃO DOS USUÁRIOS COM O SISTEMA.....     | 20        |
| 3.4.3 CASOS DE USO .....                                     | 21        |
| 3.5 ARQUITETURA DO SISTEMA.....                              | 22        |
| 3.6 DESCRIÇÃO DA INTERFACE COM O SISTEMA .....               | 23        |
| 3.7 DIAGRAMA DE CLASSES E DE SEQUÊNCIA.....                  | 24        |
| 3.8 CONCLUSÃO.....   | 27        |
| <b>CAPÍTULO 4 .....</b>                                      | <b>28</b> |
| 4 PROJETO, DESENVOLVIMENTO E TESTES .....                    | 28        |
| 4.1 INTRODUÇÃO.....  | 28        |
| 4.2 ARQUITETURA.....   | 28        |
| 4.2.1 DIAGRAMA DE DISTRIBUIÇÃO E COMPONENTES .....           | 28        |
| 4.3 BASE DE DADOS.....                                       | 29        |
| 4.4 MODELO DA INTERFACE HOMEM-MÁQUINA.....                   | 31        |
| 4.4.1 USUÁRIOS DO SISTEMA .....                              | 31        |
| 4.4.2 PERFIL DOS USUÁRIOS DO SISTEMA.....                    | 31        |
| 4.4.3 MANUAL DE ESTILO PARA O PROTÓTIPO DAS TELAS .....      | 32        |
| 4.4.4 PROTÓTIPO DAS TELAS.....                               | 33        |
| 4.4.5 PROCEDIMENTOS OPERACIONAIS .....                       | 33        |
| 4.5 PLANO DE TESTES .....                                    | 34        |
| 4.5.1 DESCRIÇÃO DO PLANO DE TESTES.....                      | 35        |
| 4.5.2 ELABORAÇÃO DO PLANO DE TESTES .....                    | 35        |
| 4.5.2.1 SELEÇÃO E ESPECIFICAÇÃO DOS MÉTODOS ESSENCIAIS ..... | 35        |
| 4.5.2.2 ESPECIFICAÇÃO DOS CASOS DE TESTE .....               | 36        |

|  |            |
|--|------------|
| 4.6 CODIFICAÇÃO, TESTES E IMPLANTAÇÃO .....                                      | 36         |
| 4.7 EXTENSIBILIDADE DO SISTEMA .....   | 37         |
| 4.8 ADERÊNCIA DO TRABALHO AOS OBJETIVOS PROPOSTOS. ....                          | 38         |
| 4.9 CONCLUSÃO.....   | 39         |
| <b>CAPÍTULO 5 .....</b>  | <b>40</b>  |
| 5 CONCLUSÃO .....  | 40         |
| 5.1 RESUMO.....  | 40         |
| 5.2 ANÁLISE DE RESULTADOS E COMENTÁRIOS.....                                     | 41         |
| 5.3 CONTRIBUIÇÕES .....  | 43         |
| 5.4. SUGESTÃO PARA FUTUROS TRABALHOS .....                                       | 44         |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>  | <b>46</b>  |
| <b>REFERÊNCIAS CONSULTADAS .....</b>   | <b>51</b>  |
| <b>GLOSSÁRIO.....</b>  | <b>57</b>  |
| <b>APÊNDICE A – DIAGRAMAS DE CASO DE USO.....</b>                                | <b>58</b>  |
| <b>APÊNDICE B – DESCRIÇÃO DOS ATRIBUTOS DAS CLASSES.....</b>                     | <b>73</b>  |
| <b>APÊNDICE C – DESCRIÇÃO DOS MÉTODOS DAS CLASSES.....</b>                       | <b>78</b>  |
| <b>APÊNDICE D – DIAGRAMAS DE SEQÜÊNCIA.....</b>                                  | <b>80</b>  |
| <b>APÊNDICE E – PROTÓTIPO DAS TELAS DO GERENCIADOR DE SERVIÇOS....</b>           | <b>89</b>  |
| E.1 INTRODUÇÃO .....   | 89         |
| E.2 TELA PRINCIPAL DO SISTEMA .....  | 89         |
| E.3 TELA: CADASTRO DE SERVIÇOS.....  | 91         |
| E.4 TELA: CADASTRO DA INTERFACE DO SERVIÇO.....                                  | 94         |
| E.5 TELA: CADASTRO DOS COMPONENTES UTILIZADOS PELO SERVIÇO .....                 | 95         |
| E.6 TELA: CADASTRO DOS PROBLEMAS COM O SERVIÇO .....                             | 97         |
| E.7 TELA: ESTORNO DAS ÚLTIMAS MODIFICAÇÕES .....                                 | 98         |
| E.8 TELA: ALTERA ESTADO DO SERVIÇO .....   | 100        |
| E.9 TELA: CADASTRO DOS RESPONSÁVEIS.....   | 102        |
| E.10 CONCLUSÃO .....   | 103        |
| <b>APÊNDICE F –PROJETO DOS TESTES DE INTEGRAÇÃO – CONTRATOS DE CLASSES. ....</b> | <b>104</b> |
| <b>APÊNDICE G –DETALHES DA ESPECIFICAÇÃO DOS CASOS DE TESTE.....</b>             | <b>110</b> |

# CAPÍTULO 1

## 1 INTRODUÇÃO

### 1.1 Motivação

Os servidores de aplicação (*Application Servers* – ASs) nos últimos anos se tornaram muito populares, provendo uma plataforma eficiente e robusta para a execução de aplicativos residentes nos próprios servidores, genericamente denominados de serviços, atendendo atualmente ao público de Internet, além da própria rede local. Tais serviços são relativos a diversos tipos de ambientes como os voltados à *WEB*, a exemplo C2B (*client to business*), B2B (*business to business*) e A2A (*application to application*), ou os que utilizam a arquitetura cliente – servidor de três ou mais camadas.

Serviço é uma aplicação que contém a lógica do negócio (ou lógica da aplicação), disponibilizado através de contrato de uso que se estabelece entre o aplicativo cliente e o próprio serviço, por meio do uso das interfaces de programação. O contrato de uso descreve o que a interface compromete-se a realizar, pois formaliza as implicações impostas pelos requisitos (ALMEIDA, 2001). O serviço reside nos servidores de aplicação e é executado em decorrência de requisições efetuadas pelos diversos aplicativos clientes (AHOKAS, 2003; TROFIN, 2004a).

A figura 1.1 mostra a arquitetura cliente - servidor de três camadas, em particular o servidor de aplicação que abriga os serviços, sendo que existem serviços que podem utilizar outra camada, como por exemplo, a de dados. Dentre esses provedores de dados é possível citar os gerenciadores de banco de dados relacionais e os servidores de arquivos. Finalmente, esta tríade de camadas se torna completa com a camada cliente onde estão situados os aplicativos consumidores dos serviços.

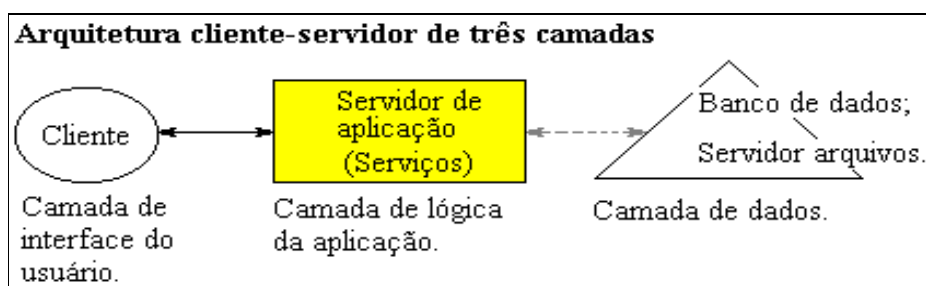


Figura 1.1 – Arquitetura cliente – servidor de três camadas.

Os serviços são instalados em máquinas robustas<sup>1</sup>, os servidores de aplicação, através de recursos administrativos disponibilizados pelos próprios servidores. Esses recursos são o cadastro dos serviços, os controles operacionais de ativação e desativação dos serviços, os mecanismos de segurança, a monitoração de desempenho e informações estatísticas, tais como o número de transações com falhas e com sucesso, além da consulta do estado dos serviços.

<sup>1</sup> Máquinas robustas implica neste contexto em grande quantidade de processamento, redundância nos armazenamentos de disco e também redundância em alguns outros itens do *hardware*.

As principais empresas fornecedoras de servidores de aplicação são: BEA, Borland, IBM, Microsoft, Oracle, SAP e SUN com os respectivos produtos: WebLogic Server Premium e Tuxedo; Borland Enterprise Server, AppServer Edition; WebSphere Enterprise CICS/IMS; MSAP - Microsoft Application Platform - .NET Framework e COM+; Oracle9iAS J2EE e JSP/Servlet Containers; SAP Web Application Server e o Sun ONE Application Server Enterprise (MCCOY; FEIMAN, 2003, 2004b; MOHAN, 2001, 2002, NATIS, 2005).

Entretanto, analisando os recursos administrativos de alguns dos servidores de aplicação (BEA WebLogic Server 7.0, 2002; IBM, 2001; Borland, 2005a; Microsoft; JBoss Application Server 4.0; SYSTINET, 2002; KRILL, 2004) nota-se que não há uma divulgação clara dos recursos administrativos ou algum desses são limitados ou, até mesmo, inexistentes.

No caso da não divulgação clara dos recursos administrativos mesmo em manuais dos produtos, pesquisas em diversos documentos (manuais operacionais, artigos de marketing do produto e artigos administrativos de *forum*) e testes funcionais revelam a existência dos mesmos recursos básicos de administração apresentados acima: cadastramento, funções de ativação e desativação de serviços, mecanismos de segurança e consulta a informações estatísticas. Os recursos apontados são insuficientes para uma administração otimizada, efetiva e pró-ativa, pois não geram subsídios detalhados de uso, controle, eficiência e qualidade que antecipem futuras dificuldades ou que forneçam diagnósticos precisos sobre o que ocorre com os serviços, como por exemplo utilizar informações históricas de outras instalações de serviços para prever problemas de conflitos de instalação com outros componentes. Sobre a qualidade é importante destacar alguns critérios para o contexto de administração, tais como, as características de manutenibilidade (esforço necessário para localizar e consertar uma falha no programa dentro do seu ambiente operacional), flexibilidade (facilidade de realizar modificações necessárias no ambiente operacional), portabilidade (esforço necessário para transferir um programa de um ambiente para outro) e reusabilidade (facilidade de reusar o *software* em um contexto diferente) (PRESSMAN, 2001). Esses critérios de qualidade podem até estar embutidos nos serviços, mas precisam estar registrados para viabilizar a administração do ambiente e dos próprios serviços.

São limitados os recursos administrativos relativos à captura e controle de informações a respeito dos serviços (entre as quais as especificações, os problemas, soluções e as modificações que sofrem) e à capacidade de dispor destas informações para a elaboração de diversos relatórios operacionais e gerenciais. Esses recursos são necessários para uma gestão fundamentada, otimizada, efetiva, confiável, eficiente e pró-ativa, como por exemplo, a rastreabilidade das modificações (ou dos problemas e das soluções) dos serviços, buscando otimizações para a eficiência dos processos tendo em vista a qualidade dos serviços ofertados (*Quality of Service – QoS*) ou o controle dos problemas pendentes e de suas soluções.

Freqüentemente, o gerenciamento dos serviços provido pelos servidores de aplicação depende de ferramentas externas (providas pelo sistema operacional ou adquiridas de fornecedores). O gerenciamento acaba sofrendo limitação em virtude da utilização dessas ferramentas não integradas ao ambiente dos servidores de aplicação, o que implica em perda para a gestão da informação e de controle, e para a automatização do serviço. De acordo com Feiman (FEIMAN, 2004b), a justificativa da atual dependência destas ferramentas externas está na atual estratégia de vendas (fragmentação dos produtos) dos servidores de aplicação,

entretanto os fornecedores estão tendenciosos para o desenvolvimento de grandes *frameworks* proprietários e integrados.

Portanto, para gerenciar melhor os serviços ofertados solucionando as limitações administrativas há necessidade do levantamento, do registro e da gestão de várias informações a seu respeito, tais como, aspectos próprios, ambiente onde se inserem, informações sobre suas dependências com o ambiente, critérios de uso, registro e acompanhamento de problemas e de soluções e o histórico das modificações ocorridas com o ambiente e com os serviços (LARSSON, 2000; ARSANJANI, 2002).

São inexistentes, do ponto de vista administrativo dos serviços, o registro e o acesso às informações para reuso, levantamento e controle dos responsáveis pela homologação e pelo desenvolvimento dos serviços, controle dos contratos dos serviços, especificação detalhada das interfaces, controle dos componentes usados e gestão de configurações e versões. Esses recursos são importantes para reaproveitamento dos serviços existentes (diminuição de custos dos projetos), controle, registro histórico, acionamento dos usuários responsáveis, análises de impacto de alterações, eliminações e conflitos (diminuição de custos de suporte).

Portanto, para criar os recursos inexistentes e gerenciar melhor os serviços e seus responsáveis deve-se atender às questões apresentadas anteriormente.

A questão de gerenciamento de serviços é preocupação bastante atual de uma comunidade internacional e tem sido abordada em congressos internacionais tais como o SIGMOD (*Special interest group on management of data*) de 2002 (MOHAN, 2001, 2002).

## 1.2 Objetivo

O objetivo desse trabalho é a especificação e projeto de extensão das funções gerenciais de um servidor de aplicação incorporando ao mesmo recursos para a gestão dos diversos serviços ofertados. No âmbito gestão, o projeto aplicará as práticas da disciplina de gerência de configuração e versionamento e as práticas de gerência de modificações (ou alterações) para registrar e controlar o ciclo de vida dos serviços no ambiente de produção. Sendo assim, esse gerenciador de serviços disponibilizará um ambiente interativo que viabilizará ao gestor atuar nas tarefas de documentação (registro) dos aspectos de ambiente e dos serviços, registro e acompanhamento de problemas e soluções ocorridos com os serviços, implantação e re-implantações dos serviços no ambiente de produção e desativação permanente dos serviços. O sistema está fundamentado em um repositório de informações que é disponibilizado para leitura das análises operacionais e gerenciais e, além disso, por automatizações internas, o sistema divulgará o acervo dos serviços para o público desenvolvedor de aplicativos com o intuito de manter todos os envolvidos atualizados e conscientizados, estimulando o reuso dos serviços.

Os recursos funcionais propostos a partir da elaboração, integração e extensão dos recursos atuais permitem:

- Estabelecer e aplicar procedimentos de implantação de serviços novos e re-implantação de serviços existentes que foram modificados.
- Estabelecer e aplicar procedimentos para a desativação permanente dos serviços.

- Estabelecer e aplicar procedimentos de registro de problemas com as suas posteriores soluções, ocorridos com os serviços.
- Gestão das informações sobre o acervo de serviços.
- Gestão de configurações e versões de serviços.

Estas funcionalidades se propõem a enriquecer a documentação dos serviços, registrando em um banco de dados de informações, o acervo, os problemas, as soluções dos problemas e todas as modificações que ocorrem com os serviços durante o seu ciclo de vida, dentro de um ambiente de produção. A recuperação das informações contidas no banco permite desde a criação de consultas rotineiras até análises operacionais, gerenciais, históricas, de rastreabilidade, de impactos, de anomalias, de tendências e de previsões. O que se propõe é criar uma infra-estrutura independente, que através dos registros operacionais viabilize o apoio à tomada de decisão sobre os serviços ofertados, vislumbrando confiabilidade, eficiência, disponibilidade e qualidade dos mesmos.

### **1.3 Resultados esperados e contribuições**

O resultado esperado do trabalho é uma especificação detalhada de todos os requisitos necessários para a elaboração do projeto, uma especificação do projeto focando as funcionalidades e o desenvolvimento das funcionalidades mais críticas e/ou específicas em forma de um protótipo. Portanto, para compor a análise constarão no protótipo as funcionalidades que viabilizam a gestão das informações sobre o acervo de serviços e a gestão de configurações e versões dos mesmos. Estas etapas acrescidas de uma análise geral viabilizarão a conclusão final do trabalho.

O trabalho em questão contribui com uma especificação de infra-estrutura para a gerência de serviços que pode atender aos objetivos propostos de forma modular, independente e expansível. Através da metodologia orientada a objetos (OO), que rege o desenvolvimento do projeto, é possível integrar por meio de interfaces essa infra-estrutura com os atuais servidores de aplicação ofertados pelas plataformas .Net e J2EE.

### **1.4 Metodologia**

O desenvolvimento do trabalho se inicia com a análise dos documentos publicados na área de servidores de aplicações, serviços e sua gerência. As fontes de informação são livros, revistas, artigos, dissertações, teses, *sites*, palestras, conferências e as documentações de produtos.

A seguir, é feita a análise dos requisitos funcionais e não funcionais para atender à necessidade da gerência de serviços sobre os aspectos de documentação (registro), configuração, versionamento, modificação, problemas, soluções e reuso.

Com embasamento nas informações das pesquisas e dos requisitos é feita a análise de alguns dos ambientes atuais dos servidores de aplicações (BEA WebLogic Server 7.0, 2002; JBoss Application Server 4.0; SYSTINET, 2002; IBM, 2001; KRILL, 2004; Microsoft) em busca das funcionalidades que operacionalizam os recursos administrativos e gerenciais e seus respectivos limites.



Em seguida, esses recursos são apresentados e questionados à luz dos requisitos, o que motiva a proposta do trabalho para extensão dos mesmos, focando-se no gerenciamento de aspectos e características básicas, de configuração, de versionamento, de modificações, de problemas e soluções dos serviços, além da necessidade de gerir o acervo dos serviços que estimula o reuso.

Para realizar a extensão dos recursos o material pesquisado e analisado é utilizado na elaboração da especificação do projeto de extensão dos recursos gerenciais dos servidores de aplicação. Nas fases de especificação constam as seguintes elaborações: levantamento dos requisitos funcionais e não funcionais; criação do cenário de uso do gerenciador de serviços (apresenta as informações coletadas na fase de levantamento de requisitos na forma de casos de uso); criação do diagrama de sequência (apresenta a interação entre os objetos das classes envolvidas) e a criação do diagrama de classes (descreve o domínio de informação do *software*).

O próximo passo é a especificação do projeto, o que requer uma descrição da arquitetura, das estruturas de dados, dos módulos do programa, da interface com o usuário e da estratégia de teste.

Em posse da especificação do projeto é construído um protótipo do gerenciador de serviços para funcionar em uma máquina Intel com a plataforma Microsoft. Esse ambiente possui as seguintes características: uma CPU com 2.6 Ghz, 512 Mbytes de memória, sistema operacional Windows Server, gerenciador de banco de dados relacional *Microsoft SQL Server 2000* e o servidor de aplicação da Microsoft. O desenvolvimento do protótipo é feito com a ferramenta *Microsoft Visual Basic*.

Em seguida, são aplicados os testes funcionais sobre o protótipo que permite verificar a aderência das funcionalidades às especificações. O resultado desses testes permite apresentar os benefícios trazidos para uma gestão mais fundamentada e efetiva, quando comparadas às funcionalidades dos sistemas atuais.

## 1.5 Organização do Trabalho

O trabalho está organizado em cinco capítulos e diversos apêndices, como segue.

- Capítulo 1 – Introdução.

Apresenta a motivação, o objetivo, as contribuições esperadas, a metodologia e a organização do trabalho.

- Capítulo 2 – Conceitos principais e estado da arte.

Apresenta o material utilizado na pesquisa (artigos, livros, revistas, teses, dissertações, *sites*, palestras, documentos de congressos, manuais técnicos, entre outros.) de forma sintetizada, integrada e alinhada ao objetivo do trabalho, possibilitando sustentar o desenvolvimento do mesmo. Esse capítulo discorre sobre Serviços, Servidor de aplicação e Gerenciamento de serviços.

- Capítulo 3 – Levantamento, análise e especificação de requisitos.

O capítulo trata do levantamento de requisitos (funcionais e não funcionais), o que permite uma compreensão global do domínio de informação tratado pelo sistema, tanto quanto do seu escopo. O capítulo aborda os resultados da análise desses requisitos no formato dos diagramas de caso de uso, explora os casos de uso nos diagramas de sequência em busca de classes e de funcionalidades adicionais, viabilizando especificar todas as informações analisadas nos diagramas de classes. Portanto, os resultados dos levantamentos e das análises provêm as informações necessárias para a especificação da arquitetura do sistema e da interface de relacionamento deste com o servidor de aplicação.

- Capítulo 4 – Projeto, desenvolvimento e testes.

Neste capítulo apresenta-se o detalhamento de cada uma das funções especificadas no projeto, bem como é proposto o plano de testes para verificar a aderência às especificações. Os resultados dos testes também são apresentados.

- Capítulo 5 – Conclusão.

Este capítulo apresenta um breve resumo do trabalho e analisa os resultados obtidos quanto à aderência aos requisitos iniciais e quanto aos testes das principais funcionalidades. São, também, descritas as possíveis restrições do projeto. O capítulo discorre sobre as contribuições do trabalho e sugere temas para futuras pesquisas.

- Apêndices

Os apêndices são partes integrantes e vitais para o trabalho, pois contém os detalhes de cada assunto abordado. Neles estão contidas especificações, diagramas e protótipos das telas de interface, além de diversas figuras e tabelas.

Participam do trabalho os apêndices:

- Apêndice A – Diagramas de Caso de Uso.
- Apêndice B – Descrição dos atributos das classes.
- Apêndice C – Descrição dos métodos das classes.
- Apêndice D – Diagramas de Sequência.
- Apêndice E – Protótipo das telas do Gerenciador de Serviços.
- Apêndice F – Projeto dos testes de integração – Contratos de Classes.
- Apêndice G – Detalhes da especificação dos casos de teste.

## CAPÍTULO 2

### 2 CONCEITOS PRINCIPAIS E ESTADO DA ARTE

#### 2.1 Introdução

Esse capítulo apresenta uma síntese dos conceitos e definições que envolvem a área de servidores de aplicações bem como serviços e sua gerência, necessários aos objetivos do trabalho.

#### 2.2 Serviços

Para responder de forma rápida às alterações das condições dos negócios as empresas precisam ter capacidade de alterar rapidamente os processos de negócio e os sistemas. Em virtude dessas necessidades os arquitetos e desenvolvedores estão utilizando componentes de negócio, ou melhor dizendo, serviços, pois eles detêm as características de serem granulares, independentes, acopláveis, colaborativos e reutilizáveis. Essa ação de criar serviços é justificada pela análise de produtividade da *Gartner Group*, que afirma que em oito anos o aumento na produção de *softwares* passou de 300 para 420 pontos de função, refletindo um crescimento anual de 3% a 4%, ou seja, um crescimento baixo para as necessidades de respostas dinâmicas das empresas aos negócios, mas esses valores indicam o máximo da capacidade humana de escrever código (FEIMAN, 2005b). Mediante a esses valores e a necessidade do dinamismo, a solução para atender às necessidades de uma empresa em tempo real é empregar esses mesmos recursos na elaboração, registro e utilização de serviços e componentes reutilizáveis.

Serviço é um processo comercial reutilizável (HOTLE, 2004). Tecnicamente um serviço pode ser definido como um recurso que contém a lógica de negócio que expõe suas funcionalidades através de uma interface previamente programada e claramente identificada, o que significa que é desenvolvido para ser consumido preferencialmente por aplicativos ou por outros serviços. O método de invocá-lo e os possíveis resultados da invocação são descritos por um contrato (SYSTINET, 2002; MCCOY, FEIMAN, 2003; FEIMAN, 2004).

Os serviços se comportam como uma “caixa preta”, pois seus códigos internos são independentes da natureza e do propósito dos requisitantes. Os serviços geralmente são formados pela aglomeração de componentes mais simples (componentes orientados a objetos) como os .NET ou classes Java, uma vez que essa composição estabelece o processo do negócio (MCCOY, FEIMAN, 2003).

Atualmente os serviços são desenvolvidos segundo a tecnologia orientada a objetos, entretanto, a construção desses pode ser feita ou não com esta tecnologia (COYLE, 2001). Essa forma de construir serviços permite que rotinas advindas de outros paradigmas de programação, como por exemplo, a programação estruturada, possam ser adaptadas (*wrapped*) e reutilizadas pela tecnologia de serviços (COYLE, 2001; MCCOY, FEIMAN, 2003; LHEUREUX, 2003). A tecnologia de adaptação (*wrappers*), comercialmente ou popularmente reconhecida como “aplicativos compostos”, consiste na elaboração de um programa que traduz os dados e o comportamento do que será adaptado em interface padrão (LHEUREUX, 2003). Esse é um recurso importantíssimo, pois permite buscar na própria empresa funcionalidades comuns de sub processos já criados, validados e com custo de desenvolvimento amortizado para reutilizá-los como serviços.

Com a oferta de serviços na Internet, passa a existir um aumento do consumo e, conseqüentemente, também da diversidade e da colaboratividade dos serviços. Sendo assim, a previsão de consumo de serviços para 2008 é de que mais de 75% das novas aplicações serão construídas com serviços (HOTLE, 2004b).

Para atender a essas novas necessidades da Internet foi desenvolvida a tecnologia dos *Web Services* com os seus respectivos padrões, que oferece para os *sites* as funcionalidades dos serviços através dos servidores de aplicação que gerenciam-nas e oferecem-nas com múltiplos instanciamentos, com simultaneidade e com alta disponibilidade. Esse provimento atende a diversos aplicativos clientes locais, remotos e também atende as plataformas dos *Web Services*.

Um dos padrões advindos com o *Web Services*, a linguagem de definição de serviço (*Web Service Definition Language* – WSDL), contribui para esse trabalho com os conceitos de como descrever os serviços. Esse padrão permite descrever em detalhes o contrato que existe entre o cliente (aplicativo cliente) e o serviço a ser requisitado (SYSTINET, 2002; BURBECK, 2000; BOOTH, 2004). O contrato é elaborado por uma linguagem neutra, clara, padronizável e extensível, o XML (*eXtensible Markup Language*) e apresenta as especificações das propriedades de serviço, tais como, o que o serviço faz, onde está localizado, como invocá-lo, entre outros. (GLASS, 2001; BAX, LEAL, 2001; SYSTINET, 2002). Além dessas informações, a extensibilidade do WSDL (provida pelo XML) permite que seja incorporada à especificação do serviço os requisitos não funcionais, tais como os custos dos préstimos, importantes para garantir a troca de informações necessárias para o consumo de serviço e questões contratuais, entre outras (BURBECK, 2000).

Outro padrão utilizado pelos *Web Services*, que contribui para esse trabalho com idéias de catálogo, armazenamento e recuperação de serviços é o UDDI – *Universal Description, Discovery and Integration*, popularmente conhecido como as “páginas amarelas dos serviços” (WORTHEN, 2002). Este padrão permite publicações e pesquisas de serviços (BAX, LEAL, 2001; AREVOLO, 2003). O UDDI utiliza um formato padronizado para o armazenamento e para a recuperação das informações a respeito dos serviços, dos provedores dos serviços e das definições técnicas das interfaces. Desta maneira, esse padrão é muito útil para pesquisas dinâmicas de serviços no âmbito restrito de uma empresa ou na vastidão da Internet (SKONNARD, 2002).

Quanto à questão de reuso dos serviços em nível organizacional, esse trará vários benefícios, tais como redução do tempo de desenvolvimento de um projeto, redução do custo de desenvolvimento, redução do tempo de liberação do projeto para o marketing, redução do esforço em testes, redução do custo de suporte e economia de recursos (tais como o armazenamento e administração). Portanto, para atender à questão de reuso, os benefícios exigem custos altos com a criação e manutenção de uma biblioteca de serviços, com o levantamento e o cadastramento do inventário dos serviços, com as formas de busca no acervo da biblioteca, com a divulgação constante do acervo da biblioteca, com a gestão dos versionamentos dos serviços, com a capacitação para utilização da biblioteca e, finalmente, com o incentivo no desenvolvimento de serviços reutilizáveis<sup>2</sup> (HOCHMULLER, MITTERMEIR, POZEWAUNIG, 2002; LIU, X. *et al.*, 1999; CRNKOVIC, LARSSON, 2001; FEIMAN, 2004, 2005, SILVA, 2003, PRESSMAN, 2001).

---

<sup>2</sup> Segundo os analistas da Gartner Group o desenvolvimento de serviços reutilizáveis consomem 20% a mais de tempo de desenvolvimento.

## 2.3 Servidor de aplicação

O servidor de aplicação é um produto de *software* baseado em componentes que residem em uma camada intermediária de uma arquitetura de três camadas (figura 1.1). Ele estabelece uma infra-estrutura que permite, entre outras funcionalidades, o cadastramento e o provimento dos recursos funcionais para os diversos aplicativos clientes ou para outros serviços. Os recursos funcionais ofertados são serviços e esses podem ser criados (desenvolvidos) ou adquiridos e adicionados ao servidor de aplicação.

Atualmente existe na área de servidores de aplicação duas grandes frentes tecnológicas, o .NET e o J2EE, portanto, exceto a Microsoft, todos os servidores de aplicação são baseados em *Java 2 Enterprise Edition* (J2EE) (FEIMAN, 2003).

Os fornecedores dos servidores de aplicação compõem dois grupos, o de *softwares* livres com o Jboss e o Jonas, e os comerciais, com a BEA WebLogic, a IBM WebSphere, o Oracle9i Application Server, a Sun ONE (iPlanet), o Sysbase EAServer, a Iona, a Borland, a Fujitsu Interstage, a Computer Associates, a Siemens, a SAP e a Microsoft .NET (PURSNANI, 2001; MCCOY, FEIMAN, 2003; MOHAN, 2002, NATIS, 2005). A figura 2.1 apresenta uma análise classificatória da *Gartner Group* dos servidores de aplicação dos principais fornecedores, destacando algumas características, tais como, de liderança e de visão futura de implementações, onde os destaques são as empresas IBM, BEA e Microsoft (NATIS, 2005).

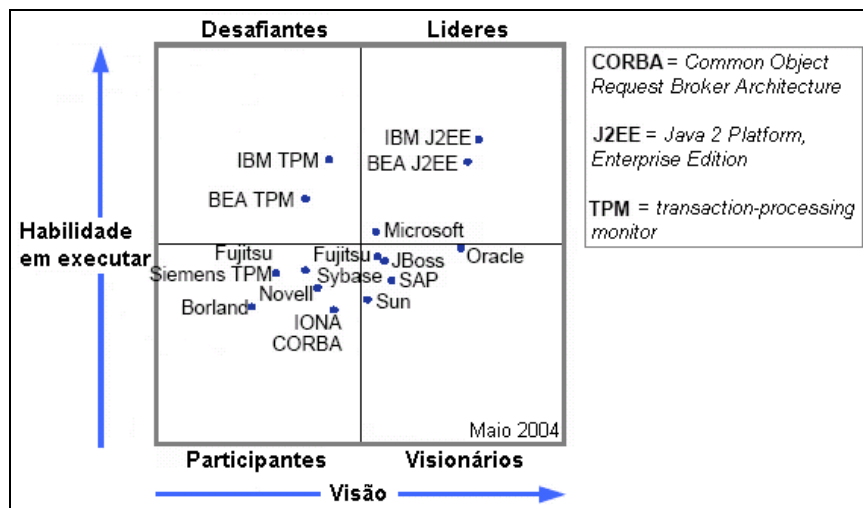


Figura 2.1 – Empresas fornecedoras de Servidores de Aplicação (NATIS, 2005).

Funcionalmente, os servidores de aplicação são equivalentes aos antigos monitores transacionais, tais como o CICS, o IMS TM, o Tuxedo, o TopEnd, o ACMS, entre outros, pois providenciam uma infra-estrutura para a execução transacional exigida pelos aplicativos, provendo características típicas como transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade), alta taxa de processos ou tarefas executadas (*throughput*), tempo de resposta menor do que poucos segundos, tráfego de mensagens síncronas e assíncronas numa rede com uma grande quantidade de clientes, gerenciamento de carga, contas, execução de processos *online*, *batch* e consultas (MOHAN, 2002; MOUGIN, BARRIOLADE, 2001).

A tecnologia dos servidores de aplicação permite que dois ou mais servidores operem em conjunto para garantir uma alta disponibilidade de suas funções (qualidade dos serviços – QoS), viabilizando assim, a clusterização, o *failover*, a capacidade de migração das conexões entre sessões e o balanceamento de carga dos serviços ofertados por esse ambiente. Além desses recursos, um único servidor de aplicação já provê a persistência das informações durante a sessão. Outras funcionalidades disponíveis nesse ambiente tratam de segurança, como a autenticação, a autorização e o *single sign-on* (MOHAN, 2002).

É importante citar que o desempenho de provimento de serviços trata da capacidade do servidor de aplicação em apresentar consistentemente a funcionalidade desejada mantendo ausência de erros, tolerância às falhas de infra-estrutura, tempo médio de execução, entre outros (INFORMATIVO, 2000).

Sendo assim, a tecnologia dos servidores de aplicação pode ser utilizada para viabilizar a integração dos sistemas e bancos de dados existentes nas empresas, no desenvolvimento de *sites* de comércio eletrônico, em portais, nos aplicativos *Web* que precisam de integração e colaboração, na infra-estrutura dos *Web Services* e na reutilização de serviços (ou componentes) (BARRY & ASSOCIATES, 2004; MCCOY, FEIMAN, 2003; FEIMAN, 2004b; TROFIN, 2004a).

Entre alguns exemplos de aplicativos que precisam das características transacionais oferecidas pelos servidores de aplicação constam os financeiros (bancos, corretoras), as seguradoras, a área da saúde, as telecomunicações, reservas (hotel, carro, aviões), controle de inventário, recursos humanos, entre outros. (MOHAN, 2002; TROFIN, 2004a).

A respeito da arquitetura, o servidor de aplicação viabiliza a expansão do modelo de arquitetura cliente – servidor de duas camadas para o modelo de três ou mais camadas. Essa expansão permite a migração da lógica de negócio do cliente e da parte servidora para os serviços localizados na camada intermediária (PURSNANI, 2001). A separação entre aplicações e serviços viabiliza o compartilhamento e o reuso dos serviços e o provimento de transparência para as aplicações cliente (FEIMAN, 2004). Essa separação permite que interfaces gráficas diferentes dos aplicativos tais como *Web*, GUI ou WAP utilizem os mesmos serviços, ou seja, que diferentes canais de negócio utilizem as mesmas regras (de negócio).

É missão do servidor de aplicação ofertar os serviços, controlar todas as solicitações dos clientes conectados e, em soluções que utilizem a terceira camada, permitir o estabelecimento de conexões com o banco de dados ou outros aplicativos. A figura 2.2 apresenta as camadas que compõem as atuais aplicações distribuídas que utilizam um servidor de aplicação. Participa da camada de apresentação uma diversidade de dispositivos clientes (*browser*, aplicativo, PDA, celular, *smart card*, quiosque) que utilizam e compartilham os serviços e os recursos providos pelo servidor de aplicação.

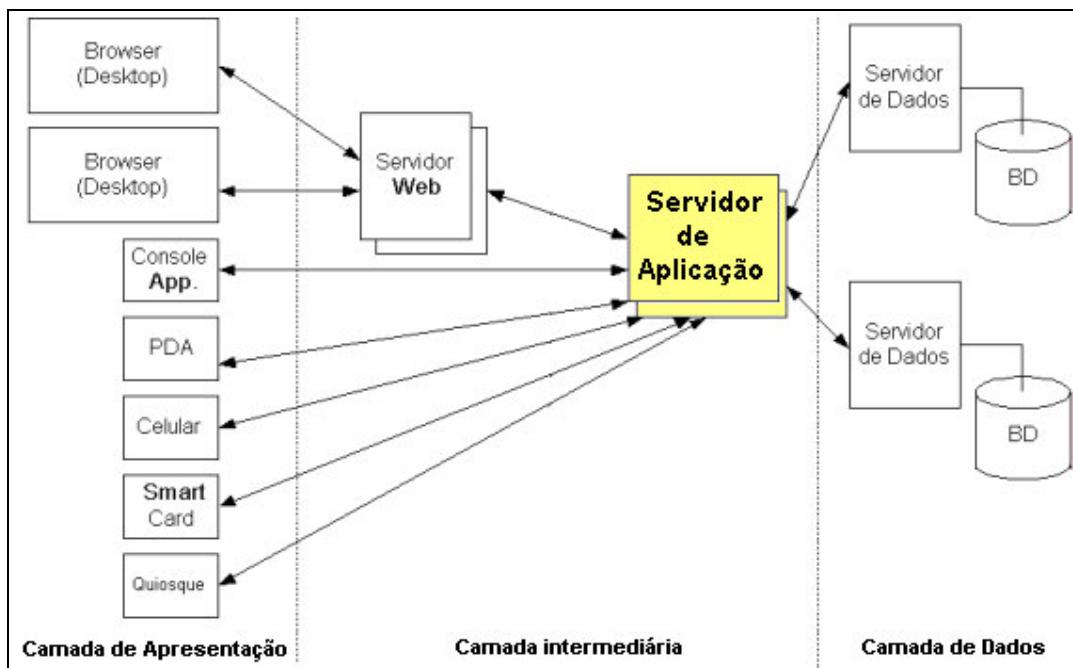


Figura 2.2 - Arquitetura distribuída em N camadas.

Assim sendo, surgem características comuns às arquiteturas multicamadas (N camadas) :

- Maior complexidade na administração do ambiente quando comparada a uma arquitetura cliente – servidor de duas camadas. Devido ao aumento dos elementos computacionais envolvidos e a sua interligação em rede, os pontos de falha aumentam e a monitoração e a depuração de problemas tornam-se mais complexas. Além disso, existe um aumento na complexidade dos aspectos de segurança.
- Apesar da independência das camadas, a instalação de novas versões de serviços pode ficar mais complexa se a aplicação ou o servidor de aplicação não tiver mecanismos que permitam determinar as janelas de horários de manutenção para essas atualizações. Corre-se o risco da perda de sessões de usuário (Usuário dos aplicativos) durante o processo de instalação.
- Os aplicativos clientes são menos complexos e monolíticos do que os seus correspondentes na arquitetura cliente-servidor de duas camadas devido ao fato que somente a lógica de apresentação permanece no cliente e o restante ter migrado para os servidores de aplicação (FEIMAN, 2004). Entretanto, com o reuso da lógica de negócio, surge a possibilidade de uma grande variedade de aplicativos clientes (GUI, WEB, WAP, entre outros.).

O ambiente do servidor de aplicação administra e gerencia os serviços ofertados que estão a seus cuidados de maneira padronizada. Ele não é responsável pelo mérito do serviço, mas por sua hospedagem e gestão, ou seja, controle, monitoramento e critérios de segurança. Além disso, o servidor de aplicação, através de regras internas, possibilita a realização de alterações dinâmicas no fluxo de execução dos serviços (processos), sem a necessidade de modificar as aplicações clientes e/ou os serviços. (FEIMAN, 2004).

Existem diferenças significativas na implementação da arquitetura dos servidores de aplicação das tecnologias .NET e J2EE. O .NET oferece uma estrutura distribuída, entretanto é menos formal e rígida do que J2EE, isto favorece a agilidade para o desenvolvimento de aplicações, mas gera dificuldades para o controle, padronizações e visibilidade dos elementos participantes de *hardware* e *software* (figura 2.2). Em contrapartida, a arquitetura dos sistemas J2EE (figura 2.3), exige mais elaboração na fase de análise, desenvolvimento e distribuição dos serviços, pois é rígida, distribuída, integrada e dotada de funcionalidades e recursos bem definidos e padronizados.

O padrão J2EE especifica como se utiliza a tecnologia *servlet* e *JavaServer Pages* (JSP) para criar páginas *Web*. A especificação J2EE define os *Enterprise JavaBeans* (EJB) que encapsulam a lógica do negócio (componentes ou serviços). Esses componentes podem se conectar a banco de dados através dos *drivers* JDBC (*Java Database Connectivity*) e podem acessar sistemas legados através da tecnologia *message-driven bean* (MDB) que utiliza *Java Message Service* (JMS). Os componentes ou serviços EJB, além de atenderem as solicitações das páginas *Web* respondem às solicitações das aplicações cliente escritas em Java que residem na camada de apresentação.

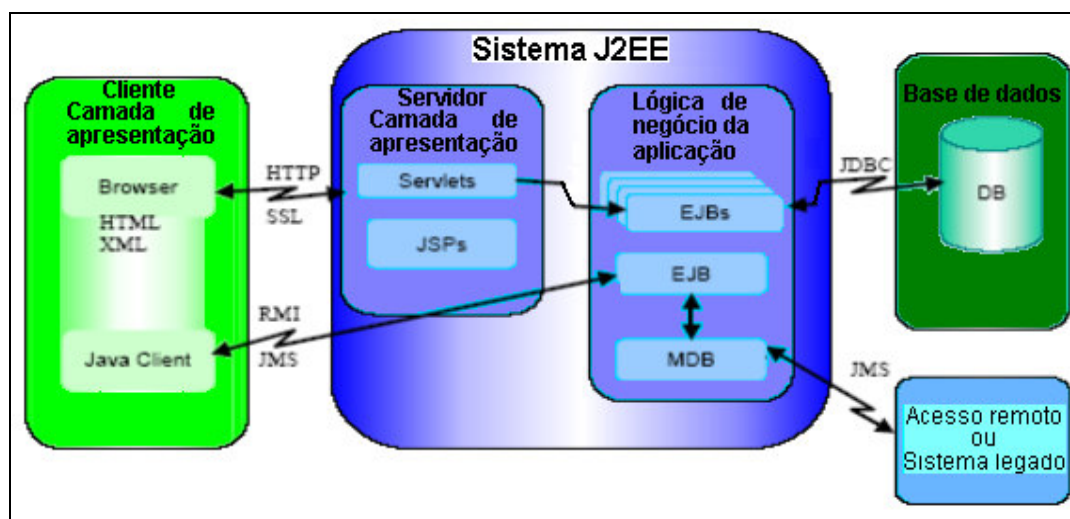


Figura 2.3 - Arquitetura dos sistemas J2EE.

Há uma dificuldade na obtenção de informações sobre as características administrativas ofertadas pelos atuais servidores de aplicação comercializados, o que preocupa tanto os fornecedores, quanto o ambiente acadêmico.

A empresa SUN (KRILL, 2004) informou amplamente e sem detalhes que desenvolverá melhorias nas características de gerenciamento e administração para a próxima versão do seu servidor de aplicação (versão 8.0). A *Systinet Corporation* (SYSTINET, 2004) anunciou em Julho de 2004 uma nova versão do seu servidor de aplicação com várias características dinâmicas, como o reuso, a configurabilidade e a adaptabilidade dos serviços.

Já, o servidor de aplicação da empresa BEA provê basicamente os seguintes recursos administrativos: implantar, remover, ativar e desativar serviços e consultar os serviços ativos



do momento. O servidor de aplicação BEA trabalha com componentes EJB (*Enterprise Java Beans*), não permite a existência de componentes com diferentes versões, pois utiliza um identificador único no registro de nome JNDI (*Java Naming and Directory Interface*), possui informações restritas sobre os serviços, não possui controle de versões ou das manutenções, mas permite a utilização da API JMX (*Java Management Extension*) para viabilizar a extensão das funcionalidades administrativas. Porém, quando é utilizado todo o *framework* de desenvolvimento BEA, esse provê o empacotamento e controle dos produtos resultado das atividades finais do ciclo de desenvolvimento do *software*.

O servidor de aplicação da IBM possui a mesma tecnologia J2EE da BEA. Isso torna ambos bem similares em recursos e restrições administrativas, entretanto o da IBM pode ter várias versões de serviços em atividade. A gestão das versões é provida por ferramentas externas – Tivoli, também de propriedade IBM (a interação Tivoli com o servidor de aplicações é realizada pela API JMX).

O histórico da Borland na questão servidor de aplicação é peculiar, pois em meados de 1998 a empresa já possuía a tecnologia Corba VisiBroker<sup>3</sup>, ou seja, um monitor transacional orientado a objetos, que disponibiliza a lógica de negócio no formato de componentes, localizados em “servidores”, para as aplicações remotas localizadas em *desktops*. Posteriormente, com o advento formal dos servidores de aplicação e com a determinação da empresa no uso da tecnologia J2EE, o Corba VisiBroker tornou-se o núcleo do servidor de aplicações Borland J2EE (BORLAND, 2005b). Em função do grande tempo no mercado do VisiBroker, de sua confiabilidade e da diversidade e grande número de clientes, a Borland criou algumas ferramentas proprietárias para executar o controle do versionamento dos serviços, gerar e recuperar pacotes de instalação e abrir e acompanhar os chamados de problemas com os serviços. Atualmente essas ferramentas estão sendo comercializadas para o gerenciamento do servidor de aplicação da empresa.

O servidor de aplicação da Microsoft possui a tecnologia .NET e basicamente provê os seguintes recursos administrativos: implantar, remover, ativar e desativar serviços e consultar os serviços ativos do momento.

No âmbito acadêmico, o pesquisador M. Trofin da Universidade de Dublin propôs, em Outubro de 2004, um *framework* em *Enterprise JavaBeans* - EJB que melhorará o desempenho dos atuais servidores de aplicação J2EE. O pesquisador descobriu um grande gasto de tempo no mecanismo de ativação de um serviço do servidor de aplicação J2EE, na tarefa de ler e reler as definições do serviço à medida que a ativação navega por seus métodos. Assim, para obter melhor desempenho propôs um *framework* capaz de remover as redundâncias de leituras (TROFIN, 2004a; TROFIN, 2004b).

De acordo com McCoy e Feiman (2003) da Gartner Group, o servidor de aplicação é a peça fundamental para as aplicações modernas, entretanto, os recursos tecnológicos para o seu gerenciamento são recentes, estão sob definição e evolução e, alguns recursos foram criados e outros adquiridos de terceiros, mas não estão totalmente bem definidos e integrados.

---

<sup>3</sup> Corba VisiBroker tecnologia mundialmente reconhecida em suas 30 milhões de licenças utilizadas em equipamentos Cisco, satélites artificiais, área bancária e etc.

## 2.4 Gerenciamento de Serviços

De uma maneira genérica, o processo de gestão ou gerenciamento compreende quatro etapas: decisão, acionamento, execução e controle. A etapa de decisão abrange a utilização de informações adequadas, a análise racional das alternativas de custo, benefício e riscos, e é finalmente concluída com as expectativas realistas dos resultados e recursos. A etapa de acionamento depende de pessoas adequadas, competentes e motivadas e também precisa de recursos, metas e controles. A etapa de controle gera benefícios como maior disponibilidade, qualidade e reatividade, viabiliza padronizações e otimizações e, embora gere custos, esses precisam ser minimizados a fim de viabilizar a gestão. Dentre estas características de controle o ideal é atuar em um sistema de controle por exceção. Essas definições de gestão colaboram na elaboração das diretrizes para o gerenciamento dos serviços.

Sendo assim, existem questões que preocupam os administradores e gestores de serviços, independentemente se o serviço é desenvolvido na própria empresa ou comprado, tais como, a documentação mais detalhada, as freqüentes modificações, as ocorrências de problemas, suas soluções e o reuso. Essas preocupações são pertinentes ao gerenciamento dos serviços e não aparecem nos atuais servidores de aplicação, entretanto há exceções pontuais, como o servidor de aplicações da BEA que possui o controle das ocorrências dos problemas.

Uma documentação mais detalhada das especificações e aspectos dos serviços ofertados viabiliza a estrutura base para todas as funcionalidades, pois é o ponto focal das consultas e atualizações dos indicadores de modificação, de problemas, de soluções e de reuso. Além disso, a habilidade de catalogar e categorizar o *portfólio* de serviços da empresa torna mais fácil o gerenciamento, ele possibilita entender o relacionamento entre serviços, os versionamentos e o gerenciamento de dependências com o ambiente e entre os serviços.

Pesquisas bibliográficas sobre gerenciamento resultaram no artigo *Web Services Architecture* (BOOTH, 2004) que acrescenta ao trabalho idéias e conceitos, sobre a definição de como é efetuado o gerenciamento de serviços em ambiente de *Web Services*. Consta nesta definição que o gerenciamento é provido através de um conjunto de funcionalidades que viabilizam o monitoramento, o controle e os relatórios sobre a qualidade e uso de serviços. A qualidade de serviço trata da disponibilidade, do desempenho e da acessibilidade. As informações de uso de serviços tratam da freqüência, da duração, do escopo e das autorizações de acesso aos mesmos.

Com o intuito de obter mais elementos que possam contribuir para uma gestão mais eficiente dos serviços, a pesquisa e a análise do trabalho buscam informações na disciplina de gerência de configuração da área de Engenharia de *Software*. Nesta disciplina, o termo “configuração” representa o conjunto de todas as características físicas e/ou funcionais do *software*. Desta forma, a configuração inclui todas as características do *software* que precisam ser controladas, ou seja, sua documentação, as versões do *software* e os documentos que identificam como as versões são modificadas, dados necessários para a operação do *software*, e qualquer outro elemento essencial do *software*.

A gerência de configuração compreende um conjunto de atividades que devem ser desenvolvidas para administrar as modificações durante o ciclo de vida do *software*. Consta na relação dessas atividades a identificação da configuração de *software* em momentos específicos no tempo, o controle das modificações de configuração, o controle das versões e manutenção da integridade e rastreabilidade do *software* (CYBIS, 2003; PRESSMAN, 2001).

As modificações são intrínsecas ao *software* e a seus produtos, e entre esses estão os serviços. As modificações dos serviços ocorrem em vários momentos do ciclo de vida e diversas são suas origens: a identificação de novos requisitos, a evolução do ambiente operacional, os avanços tecnológicos, a identificação de novas maneiras de implementar funcionalidades, necessidades de otimizações ou a mais freqüente de todas, a correção de defeitos. Portanto, são mais críticos para um ambiente de produção os resultados negativos da execução de um serviço que já sofreu modificações. Assim, esses erros e os erros originais precisam ser registrados.

A possibilidade de ocorrer modificações freqüentes dos serviços é o verdadeiro perigo para a perda de controle que, conseqüentemente, fere a gerência de configuração e, de modo mais amplo, a gerência de serviços. Atualmente, com o crescimento da complexidade do *software* e o aumento de sua granularidade, mais e mais se percebe a dificuldade crescente dos produtores de *software* em fornecer produtos íntegros em prazos razoáveis e com custos adequados. A seguir citam-se exemplos comuns de perda de controle:

- Dificuldades de integração de diversos módulos para compor o produto, por problemas de interface ocorridos por incompatibilidade ou por modificações das mesmas;
- Problemas de integridade do produto de *software* após qualquer modificação, pois erros removidos anteriormente reaparecem, ou então, funcionalidades acrescentadas desaparecem, inexplicavelmente;
- Dificuldades em identificar a última versão do produto de *software*.

Os serviços estão sujeitos a modificações por serem produtos de *software* e os servidores de aplicação estão desprovidos de controle e de administração das modificações e versionamentos.

Mas, no aspecto utilização de serviços, de acordo com Feiman (FEIMAN, 2004) da Gartner Group, a reutilização dos serviços não é uma questão tecnológica, mas sim, uma questão administrativa, pois historicamente, segundo Licht (LICHT, 2004), a gestão dos projetos ligados ao gerenciamento de serviços é realizada de forma desconexa e atemporal. Portanto, para ocorrer o reuso é necessário um repositório que funcione como uma biblioteca contendo padrões, projetos e artefatos, além de uma constante avaliação de uso, tornando assim, a solução atualizada com o que ocorre com os serviços. Outra ação necessária para o reuso é uma ampla e freqüente divulgação do acervo da biblioteca.

## 2.5 Conclusão

A síntese dos conceitos, definições e análises apresentadas permite estabelecer algumas conclusões:

- É necessário conhecer todos os serviços ofertados e suas características para se fazer sua gestão, portanto é necessário investir em documentação e em processos mais alinhados e sincronizados às freqüentes modificações a que esses estão sujeitos. A retenção dessas informações pode se favorecer dos conceitos de descrição de serviços do WSDL e, para administrar as modificações durante o ciclo de vida dos serviços em produção, a gestão pode utilizar as definições da gerência de configuração e modificação.

- A gestão precisa de um “catálogo vivo” ou “páginas amarelas” dos serviços ofertados que possa ser acessado pelos administradores dos serviços para mantê-lo atualizado e pelos desenvolvedores de *software* para viabilizar consultas e, conseqüentemente, o reuso.

Esse trabalho utiliza os aspectos do WSDL que permitem descrever em detalhes os serviços e também utiliza os aspectos do UDDI que permitem armazenar e recuperar informações em um catálogo de serviços. A proposta de utilizar nesse trabalho os padrões WSDL e UDDI possibilita que, um dia, o gerenciador de serviços também possa trabalhar com a plataforma *Web Services*.

O próximo capítulo apresenta o levantamento, a análise e a especificação de requisitos que atendem à proposta de extensão das funções gerenciais dos servidores de aplicação.

## CAPÍTULO 3

### 3 LEVANTAMENTO, ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS

#### 3.1 Introdução

O projeto do gerenciador de serviços tem como objetivo estender as atuais funcionalidades de gerenciamento de serviços provida pelos servidores de aplicação (J2EE e .NET), por meio de novos recursos (gerenciamento de configuração, versão e modificação, e a divulgação do acervo dos serviços que favorece o reuso) e através da integração entre esses aplicativos. A integração é ampla, abrangente e configurável, tornando o gerenciador de serviços independente na missão de gestão, pois permite que o mesmo comunique-se e gerencie as diversas implementações dos servidores de aplicação (BORLAND, BEA, IBM, Microsoft, entre outros). Para atender aos novos recursos do projeto é necessário criar serviços de negócio dedicados às tarefas de gerenciamento, os quais depositam as informações tratadas em um repositório compartilhado, além das interfaces destinadas aos usuários para a administração e pesquisas do acervo dos serviços.

Assim, para especificar o projeto do gerenciador de serviços, esse capítulo trata do levantamento de requisitos funcionais e não funcionais. Apresenta o cenário de uso do sistema, a arquitetura do sistema proposto, e os seus diagramas de classe e seqüência.

#### 3.2 Requisitos funcionais.

Os requisitos funcionais apresentados a seguir descrevem os serviços que o projeto (sistema) deve oferecer, como o sistema deve reagir a certas entradas e como o sistema deve comportar-se em determinadas situações. Os requisitos relevantes para o projeto abordados no trabalho são divididos nos seguintes grupos: Grupo de registro, Grupo de manutenção, Grupo de controle, Grupo de alertas e o Grupo das consultas ou relatórios.

##### a) Grupo de registro

- Registrar as informações que especificam um serviço, tais como a descrição, o tipo de serviço, o projeto de que participa, a data da inclusão no registro, a data de desativação do serviço, o responsável pelo desenvolvimento do mesmo, a linguagem utilizada na programação, a plataforma operacional em que atua, o período diário previsto para sua atividade, o tempo previsto fora do ar, a localização da instalação, entre outros.
- Registrar as configurações dos serviços.
- Registrar os aspectos da homologação do serviço, tais como data e responsável pela homologação. Isso implica que implantações em ambiente de produção somente podem ocorrer para elementos homologados.
- Registrar os aspectos contratuais dos serviços adquiridos de terceiros, tais como data de vencimento do contrato, número de identificação do contrato e a razão social do fornecedor.
- Registrar o *layout* da assinatura do serviço (interface do serviço).

- Identificar e registrar os *drivers*, os componentes e os serviços de infra-estrutura utilizados pelos serviços. Essa ação viabiliza o controle minimizando e evitando conflitos e incompatibilidades entre os serviços e a infra-estrutura do ambiente, uma vez que os elementos envolvidos estão sujeitos a versionamentos.
- Registrar os usuários responsáveis pela administração, desenvolvimento, homologação e manutenção dos serviços.

b) Grupo de manutenção

- Registrar as modificações que ocorrem com os serviços (alteração das especificações e configurações, re-implantações e desativação dos serviços, entre outros).
- Registrar os períodos e motivos de indisponibilidade dos serviços.
- Registrar os problemas que ocorrem com os serviços e as respectivas soluções.
- Alterar o estado dos serviços (ativo, inativo, suspenso).
- Atualizar o servidor de aplicações com as informações do gerenciador de serviços.

c) Grupo de Controle

- Controlar o versionamento dos serviços e dos componentes.
- Controlar e administrar o estado dos serviços (ativo, inativo, suspenso).
- Controlar os problemas pendentes dos serviços.
- Controlar a data de vencimento do contrato dos serviços.
- Administrar as janelas de horários disponíveis para planejamento de manutenção de um serviço, de um conjunto de serviços ou do servidor de aplicações.

d) Grupo de alertas (geração de avisos)

- Alertar o público responsável pelo desenvolvimento dos serviços e dos aplicativos sobre as ações de inclusão, modificação e exclusão dos serviços do acervo. O intuito do alerta é informar os responsáveis imediatamente após as modificações no acervo dos serviços. Essa divulgação automática estimula o reuso do serviço e deve minimizar inconsistências, conflitos e impactos e deve favorecer um melhor aproveitamento das modificações dos serviços nos projetos atuais e nos novos.
- Alertar o responsável (desenvolvedor) quando o serviço apresenta problema.
- Alertar o responsável (administrador) pelos serviços que existem problemas pendentes.

e) Grupo das consultas ou relatórios

- Permitir o acesso e uso do repositório de informações (base de dados) do sistema para a elaboração de consultas ou relatórios operacionais e gerenciais.

A formalização, a padronização e os detalhes de cada um desses grupos de requisitos funcionais apresentados são abordados no item 3.4 – Cenário de uso e são explorados em diversas situações nos Diagramas de Caso de Uso (Apêndice A). A análise desses diagramas permite a identificação das classes de objetos que compõem o projeto, os relacionamentos envolvidos e os seus respectivos atributos e funcionalidades.

### 3.3 Requisitos não funcionais

Os requisitos não funcionais apresentados a seguir caracterizam os aspectos de qualidade do sistema, auxiliando na definição dos atributos, propriedades, comportamento e restrições do sistema. Esses requisitos são separados em duas categorias, os utilizados para o desenvolvimento do trabalho e os requisitos necessários para sustentar o uso do resultado do trabalho (sistema).

a) Requisitos não funcionais utilizados no desenvolvimento do trabalho:

- Selecionar e utilizar uma base de dados relacional que preserve a integridade das entidades, das transações e que proporcione um tempo de resposta adequado.
- Selecionar e utilizar ferramentas geradoras de consultas e/ou relatórios para gerar os diversos tipos de visões (operacionais e gerenciais), em diversos formatos, a partir da extração dos dados do banco de dados.
- Selecionar e utilizar um servidor de aplicações que abrigue e disponibilize os serviços gerados pelo projeto. O servidor deve atuar com múltiplo instanciamento dos serviços e deve proporcionar um tempo de resposta adequado (item 2.3).
- Construir todas as páginas *Web* de consultas do projeto de acordo com as normas do W3C (*World Wide Web Consortium*).
- Padronizar visualmente e funcionalmente as telas do aplicativo (cliente – servidor ou *site*) de acordo com as boas práticas de requisitos de usabilidade.
- Descobrir, selecionar e catalogar os serviços ativos que atuam nos processos da empresa. A mesma necessidade de triagem ocorre para os respectivos responsáveis por esses serviços.

b) Requisitos não funcionais necessários para sustentar o uso do resultado do trabalho (sistema). Esses não são considerados no trabalho, porque são tarefas diárias operacionais de qualquer sistema.

- Gerar um *back up* diário da base de dados.
- Acompanhar constantemente os indicadores de desempenho da base de dados.

- Gerar *back up* dos *layouts* das consultas ou relatórios em virtude da alteração dos mesmos.
- Gerar *back ups* dos serviços em virtude da alteração dos mesmos.
- Gerar *back up* do conteúdo do *site* em virtude da alteração dos seus elementos.
- Selecionar e utilizar um servidor (ou um serviço de um determinado servidor) com a capacidade de enviar os e-mails gerados pelo aplicativo.

### 3.4 Cenário de uso

O cenário de uso tem como objetivo apresentar de forma organizada as informações coletadas na fase de levantamento de requisitos e representá-las na forma de diagramas de Caso de Uso. As etapas que participam do cenário de uso são a identificação dos atores, os tipos de interação dos usuários com o sistema e os diagramas de Caso de Uso.

#### 3.4.1 Atores que participam dos Casos de Uso

O gerenciador de serviços disponibiliza as suas funcionalidades para três atores:

1. O administrador dos serviços, responsável pela implantação (instalação), re-implantações e gestão do acervo, das configurações, das versões, das modificações, dos problemas e soluções e dos aspectos referentes aos serviços e ao ambiente em que habitam. O administrador também é responsável pela gestão dos usuários responsáveis por ações no sistema.
2. O desenvolvedor de aplicativos ou serviços, responsável por pesquisar os serviços necessários, dentre aqueles disponíveis no acervo para serem utilizados em novos projetos. Também é responsável por fornecer novos serviços para o gerenciador ou atualizações (correções) sobre os existentes.
3. O sistema responsável por automatizar as ações administrativas.

#### 3.4.2 Tipos de interação dos usuários com o sistema

Esse item trata do relacionamento operacional e gerencial entre os usuários (Administrador, Desenvolvedor e Sistema) e o sistema proposto (projeto).

Os usuários com perfil de Administrador utilizam o sistema para cumprir os requisitos funcionais contemplados pelos seguintes grupos: Grupo de registro, Grupo de manutenção, Grupo de controle e Grupo de consultas ou relatórios.

Os usuários com perfil de Desenvolvedor utilizam o sistema para cumprir os requisitos funcionais do Grupo de consultas ou relatórios. Os desenvolvedores executam pesquisas sobre o acervo (base de dados) em busca de informações atualizadas para descobrir novos serviços e utilizá-los (ou reutilizá-los) em novos projetos (ou para a revisão dos projetos atuais) e para verificar se serviços existentes tiveram alterações que possam gerar conflitos.



O perfil Sistema é responsável pelo atendimento dos requisitos funcionais do Grupo de alertas.

### **3.4.3 Casos de Uso**

O Caso de Uso trata os requisitos funcionais do sistema, proporciona descrições claras da interação entre o usuário e o sistema, e proporciona a base para os testes de validação (PRESSMAN, 2001).

O resultado da análise dos requisitos funcionais é apresentado no apêndice A na forma de Casos de Uso. Consta, na especificação individual de cada caso de uso, seu título, os atores que participam do caso, os requisitos funcionais, um texto descritivo sobre o caso, informações detalhadas de como é iniciado o caso, o fluxo de eventos contemplados pelo caso e uma figura que sintetiza visualmente o caso de uso.

Os casos de uso tratados pelo trabalho estão alinhados com os requisitos funcionais e, sendo assim, são organizados em quatro grupos funcionais:

#### **1) Grupo de registro e de manutenção**

- a) Cadastramento e manutenção do acervo de serviços.
- b) Cadastramento e administração dos serviços adquiridos de terceiros.
- c) Cadastramento e manutenção das informações sobre a interface dos serviços.
- d) Cadastramento e manutenção dos componentes utilizados pelos serviços.
- e) Cadastramento e manutenção dos problemas e soluções que ocorrem com os serviços.
- f) Anulamento (ou Estorno) das últimas modificações (nos componentes utilizados, na interface e nas informações dos serviços) que ocorreram com os serviços.
- g) Alteração do estado do serviço.
- h) Cadastramento dos usuários responsáveis pelos serviços, pela homologação, pelos registros de modificações, pelos registros de problemas e soluções, pelos registros dos componentes utilizados pelos serviços e pesquisas no acervo.

#### **2) Grupo de controle**

- i) Administração da data de expiração dos serviços.
- j) Administração dos problemas não solucionados relativos aos serviços.
- k) Administração do estado dos serviços.

#### **3) Grupo de alertas (geração de avisos)**

- l) Divulgação automática de informações sobre os serviços.

#### 4) Grupo das consultas ou relatórios

m) Disponibilização da base de informações do gerenciador de serviços para pesquisas (do acervo, de configuração, de problemas, de soluções e de modificações) para finalidades operacionais, gerenciais, de rastreabilidade e reuso.

### 3.5 Arquitetura do sistema

A figura 3.1 apresenta a arquitetura de três camadas (apresentação, intermediária e dados) utilizada tanto pelos atuais Servidores de aplicação como pela proposta de projeto do Gerenciador de serviços.

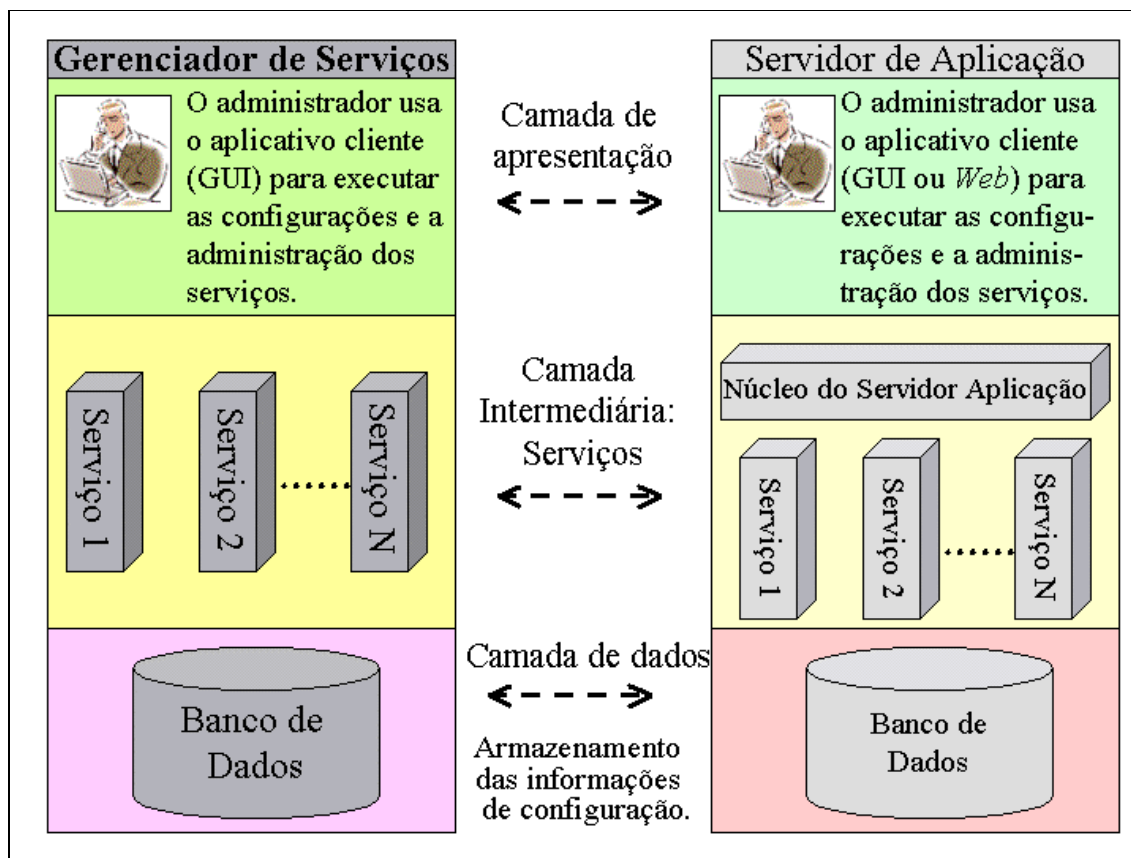


Figura 3.1 - Modelo de três camadas

A arquitetura do Gerenciador de serviços, figura 3.1 – lado esquerdo, expõe os elementos específicos (banco de dados, serviços destinados à gestão e a aplicação de gerenciamento) que provêm os recursos para as funcionalidades de gerenciamento propostas pelo projeto e as distribui e as associa em camadas (apresentação, intermediária e dados). Essa mesma similaridade de distribuição dos elementos ocorre com as funcionalidades de administração dos servidores de aplicação. A proposta do projeto é integrar o Gerenciador de serviço com o Servidor de aplicação, por meio de uma interface de programação (API) que estabelece uma comunicação direta com o núcleo do servidor de aplicação, dessa maneira a interface gráfica de administração do servidor de aplicação pode ser substituída pela GUI do gerenciador de serviços. A integração entre esses aplicativos torna-se completa quando os

serviços destinados à gestão (Gerenciador de serviços) são adicionados aos cuidados e provimentos do servidor de aplicação.

A arquitetura em uso nos Servidores de aplicação permite uma separação em camadas (três camadas: Camada de dados, Camada intermediária e Camada de apresentação) entre os dados (informações sobre as especificações e configurações dos serviços), o núcleo do servidor de aplicação (componente provedor e controlador dos serviços de negócio) e a sua interface gráfica de administração (aplicativo de administração do tipo GUI ou *browser*). Na implementação desse tipo de arquitetura, o banco de dados das configurações reside na mesma camada do núcleo (camada intermediária), porque é um banco dedicado a ele. O aplicativo de administração pode ser acessado via *Web* ou diretamente no servidor de aplicações.

A arquitetura proposta para o projeto do Gerenciador de serviços também utiliza uma separação em camadas (três camadas: Camada de dados, Camada intermediária e Camada de apresentação) entre os dados (informações dos serviços sobre as especificações, configurações, modificações, problemas, soluções e responsáveis), os serviços de gestão (controle da gestão dos serviços de negócio) e a sua interface gráfica de administração (GUI). Na implementação dessa arquitetura, o aplicativo GUI dedicado à administração deve ser instalado diretamente em estações *desktop* de gerenciamento. Entretanto, o aplicativo que viabiliza as pesquisas sobre o acervo dos serviços é disponibilizado através de *browsers* na *Web*. A segregação física do banco de dados traz como vantagem o seu compartilhamento, pois permite o acesso ao banco por outras instâncias do gerenciador de serviço e por outros aplicativos, tais como o módulo de pesquisas sobre o acervo e o visualizador de relatórios.

### 3.6 Descrição da interface com o sistema

O Gerenciador de serviços, para cumprir o objetivo de estender as funcionalidades gerenciais do Servidor de aplicação, estabelece com esse uma comunicação (interface) direta através de interfaces de programação - API (*Application programming interface*). A figura 3.2 apresenta de forma geral esses elementos envolvidos, integrados por meio de interface - API.

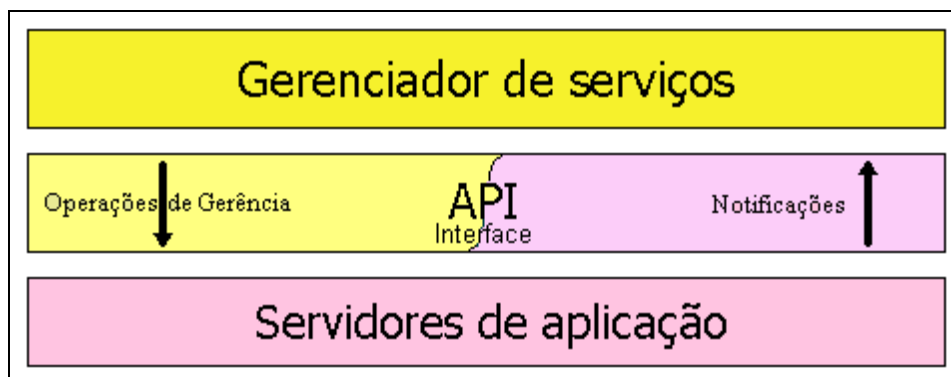


Figura 3.2 – Integração do Gerenciador de Serviços com o Servidor de Aplicações.

Dessa maneira, por intermédio de interfaces específicas (APIs), o gerenciador de serviços pode se comunicar com diferentes servidores de aplicação (IBM, BEA, Borland,

Microsoft, entre outros). Essas interfaces (APIs) são contempladas pelo projeto nas especificações das classes dos objetos (Diagrama de classes).

Pela interface (API) trafegam entre os envolvidos - Gerenciador de serviços e Servidor de aplicação - operações de gerência oriundas do sistema gerenciador (Gerenciador de serviços) e retornam notificações do sistema gerenciado (Servidor de aplicações) (SACHET, ARAKAKI, 2000) (figura 3.2 e 3.3). Todo o trâmite operacional e gerencial que se refere a questões de gerência de configuração, versionamento, especificações, modificações, problemas e soluções, são realizados pelo perfil de administrador através do gerenciador de serviços (figura 3.3).

A figura 3.3 também apresenta a comunicação que existe entre o Aplicativo Cliente (Solicitante de serviços) e o Servidor de Aplicação (Provedor de serviços: Serviço 1, ..., Serviço N). Essa comunicação é formalizada e viabilizada entre as partes por um Contrato do serviço que descreve, entre outras informações, o serviço que será fornecido com os requisitos funcionais e não funcionais, conforme descrito nos itens dos capítulos 1, item 1.1 e 2, item 2.2 (PRESSMAN, 2001).

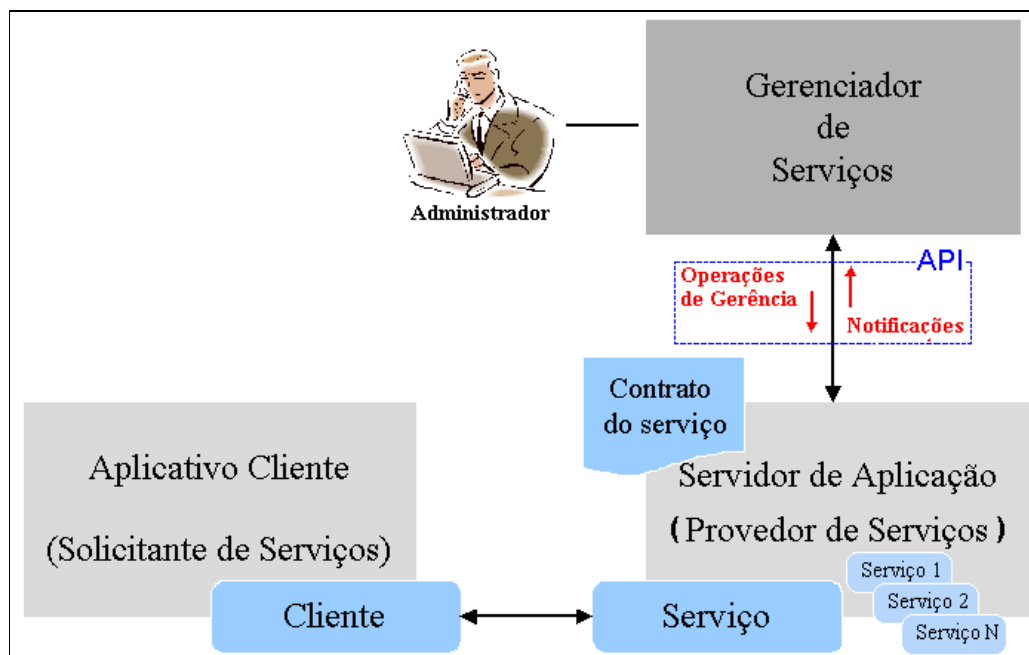


Figura 3.3 - Visão conceitual dos elementos que participam do assunto serviços.

### 3.7 Diagrama de Classes e de Sequência

Da análise de vários itens do projeto (como o objetivo, os requisitos funcionais e os diagramas de Caso de Uso) provêm informações que permitem identificar as classes, os atributos (propriedades), os relacionamentos, multiplicidades e métodos (funcionalidades).

O diagrama de classes é fundamental para a elaboração do projeto, uma vez que proporciona a identificação e organização das classes.

A figura 3.4 apresenta o diagrama de classes do projeto do gerenciador de serviços. A imagem destaca as cinco classes que compõem o sistema com os respectivos relacionamentos e suas multiplicidades. Um dos destaques do diagrama é a classe Serviço, responsável por persistir em seus objetos o acervo, as configurações e as versões dos serviços de uma empresa.

A figura 3.5 apresenta o diagrama de classes da interface (API) que viabiliza a comunicação entre o gerenciador de serviços e o servidor de aplicação (ou servidores de aplicação) conforme descrito no item 3.6. O destaque neste diagrama é o relacionamento “is-a” (“é um”) entre as sub classes e a super classe, por exemplo, a classe Borland “is-a” a classe J2EE. Através desse relacionamento ocorre a herança dos atributos e operações, o que permite a extensibilidade do sistema uma vez que viabiliza o acréscimo de novos servidores de aplicação na atual estrutura do sistema de maneira fácil e rápida.

Dessa forma, o resultado desta análise é formalizado nos diagramas de Classes apresentados pela figura 3.4 e 3.5, e em suas especificações, apêndices B – Descrição dos atributos das classes e C – Descrição dos métodos das classes.

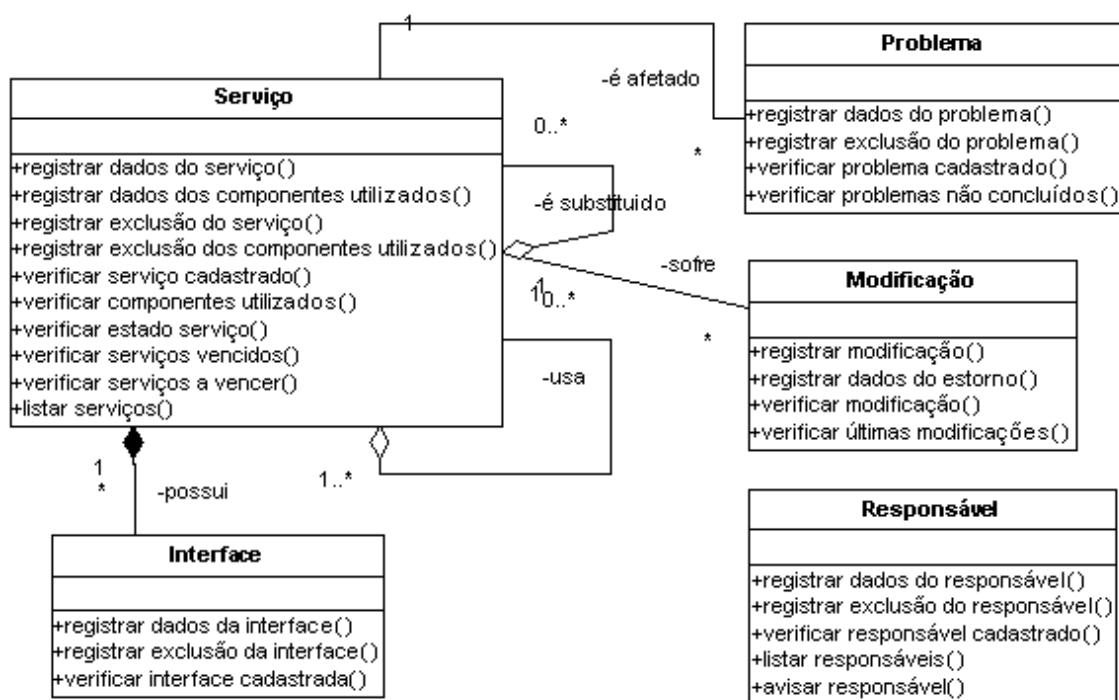


Figura 3.4 – Diagrama de Classes do Gerenciador de serviços.

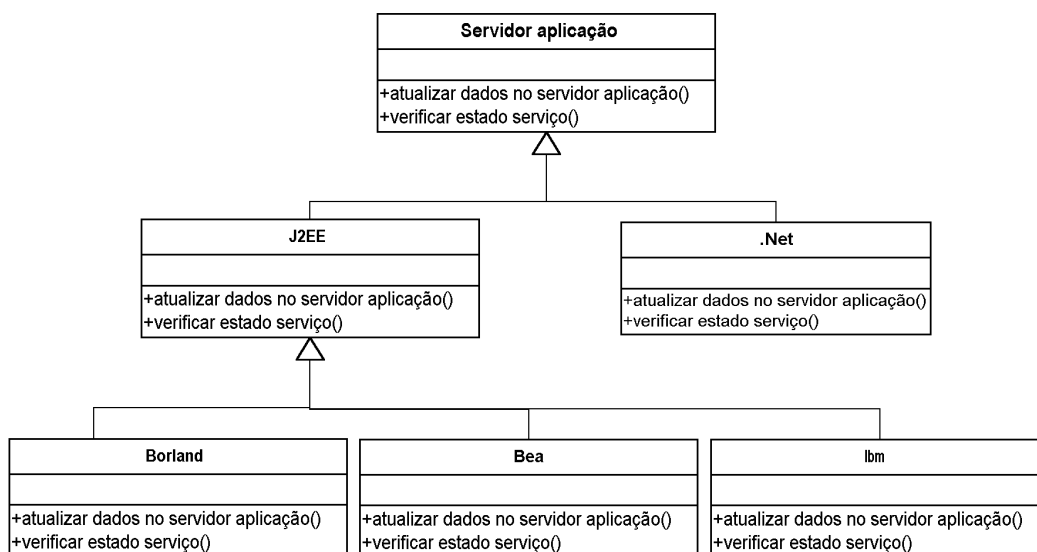


Figura 3.5 – Diagrama de Classes da API de interface entre Gerenciador de serviços e o Servidor de aplicação.

A listagem (tabela 3.1) apresenta um sumário das classes que são gerenciadas e manipuladas pelo projeto do gerenciador de serviços (figura 3.4 e 3.5).

Tabela 3.1 – Lista das Classes que participam do projeto.

| # | Classe      | Objetivo   |
|---|-------------|--|
| 1 | Serviço     | As ações que utilizam os objetos da classe Serviço permitem incluir, alterar, excluir e consultar os serviços ofertados. As persistências dos objetos desta classe representam o acervo da biblioteca de serviços. Os objetos da classe Serviço também persistem os componentes utilizados pelos próprios serviços.                      |
| 2 | Responsável | As ações que utilizam os objetos da classe Responsável permitem incluir, alterar, excluir e consultar os usuários responsáveis pelos serviços e pelas ações no sistema.  |
| 3 | Interface   | A classe trata da Interface do serviço e os seus objetos persistem a assinatura do mesmo. As ações que utilizam os objetos desta classe permitem incluir, alterar, excluir e consultar a estrutura de interface de um serviço.   |
| 4 | Modificação | O objeto da classe Modificação persiste todas as modificações que ocorrem com os serviços e com o ambiente. Esta classe é o centro nervoso do sistema, portanto os objetos dela trabalham somente com inclusões. Sendo assim, para reparar erros operacionais de registro (entre outros erros) o sistema conta com operações de estorno. |
| 5 | Problema    | O objeto da classe Problema persiste os problemas que ocorrem com os serviços. Além do registro dos problemas, também é obrigatório persistir as suas soluções.  |

Continuação da tabela 3.1

|   |                    |  |
|---|--------------------|--|
| 6 | Servidor Aplicação | A “classe Servidor Aplicação”, é uma classe abstrata que viabiliza a ligação funcional entre o sistema gerenciador de serviços e o objeto de gestão o servidor de aplicação. As sub classes .Net e J2EE pertencem a “classe Servidor Aplicação”, conseqüentemente, obtêm os métodos desta por herança. Desta forma, para atender os diversos servidores de aplicação J2EE é necessário especificar sub classes de acordo com os servidores J2EE eleitos (BEA, Borland, IBM, entre outros). |
|---|--------------------|--|

Além do diagrama de classes, os diagramas de Caso de Uso, as informações consolidadas no diagrama de Classes e as especificações dos atributos e métodos (Apêndices B e C) são os subsídios para o desenvolvimento dos diagramas de Seqüência que apresentam o comportamento colaborativo do grupo de objetos do projeto.

Os diagramas de Seqüência apresentados no apêndice D exibem nas interações entre os objetos (instâncias das classes) os cenários dos diagramas de Casos de Uso (Apêndice A), ou seja, apresentam na forma de diagramas dinâmicos a interação entre o usuário (administrador do sistema, desenvolvedor de aplicativos ou serviços, ou o próprio sistema.) e as diversas instâncias de classes (objetos) que compõem os aplicativos do sistema.

Em posse das especificações e dos diagramas acima descritos (caso de uso, classes e seqüência) torna-se possível a construção do projeto.

### 3.8 Conclusão

O capítulo em questão proporciona uma descrição geral do gerenciador de serviços apresentando os requisitos funcionais no formato dos diagramas de Caso de Uso, contextualizando os requisitos não funcionais, definindo os perfis dos usuários, atribuindo aos perfis a interação com o sistema e apresentando o modelo de seus objetos e comportamentos no formato do Diagrama de classes (figura 3.4 e 3.5).

Um aprofundamento funcional detalhado é obtido com a análise dos Diagramas de Seqüência (Apêndice D) que identificam em minúcias as operações necessárias de cada classe. Essas informações também participam do Diagrama de Classes apresentado na figura 3.4.

O Diagrama de Classes é classificado pela *Unified Modeling Language* (UML) como um diagrama estático, mas contém além das classes e relacionamentos, todos os atributos e funcionalidades que viabilizam a construção do sistema. Assim, em posse desse diagrama e dos diagramas dinâmicos (diagramas de seqüência que destaca o relacionamento entre os objetos), torna-se viável a elaboração do projeto.

Com base nos levantamentos, análise e especificações contidas nesse capítulo é possível seguir com o desenvolvimento do projeto, atuando na elaboração do protótipo das telas do gerenciador de serviços.

## CAPÍTULO 4

### 4 PROJETO, DESENVOLVIMENTO E TESTES

#### 4.1 Introdução

Esse capítulo apresenta: a Arquitetura, a Base de dados, o Modelo da Interface Homem-Máquina (IHM) e o Plano de testes, baseados nos requisitos e especificações do capítulo anterior.

#### 4.2 Arquitetura

Conforme apresentada na figura 3.1, a arquitetura do gerenciador de serviços é de três camadas, implementada em três computadores (Servidor de banco de dados, Servidor de aplicação e *Web*, e *Desktop* de gerenciamento) com as respectivas funcionalidades de armazenar e recuperar dados, prover serviços e *Web sites*, e executar os aplicativos GUI e *Web* (respectivamente gestão e *site* de pesquisas).

A distribuição da instalação e a configuração das funcionalidades em computadores dedicados a esses objetivos possibilitam uma série de vantagens: alto desempenho dos processos (o ambiente de *hardware* e *software* é customizado de acordo com o processo); maior mobilidade dos elementos instalados nos computadores (a dedicação e a especialização dos ambientes favorecem as ações de mobilidade dos elementos, pois diminui a variedade, a acoplabilidade e as relações de dependência entre os elementos e esses com o ambiente); minimização de conflitos com outros aplicativos e componentes (a variedade dos aplicativos ou componentes é menor, conseqüentemente, minimiza os conflitos); maximização nas configurações de segurança (a variedade menor dos aplicativos, geralmente, viabiliza maiores restrições de segurança).

##### 4.2.1 Diagrama de Distribuição e Componentes

O diagrama de distribuição apresenta o relacionamento entre os componentes de *software* e *hardware*, enquanto que o diagrama de componentes apresenta os vários componentes do sistema e suas relações de dependência (FOWLER, 2000). O projeto apresenta em um único diagrama, na figura 4.1, a riqueza das informações destes dois diagramas, consolidando nessa visão os computadores, as conexões, os componentes de *software* instalados nos computadores e as dependências entre os componentes.

O sistema é composto de três subsistemas: Aplicativo gerenciador de serviços, Visualizador dos relatórios ou consultas e *Web Site* para pesquisas do acervo. Os dois primeiros subsistemas são instalados nos *desktops* (estações de gerenciamento) e o terceiro em um servidor *Web*. Os subsistemas aplicativo gerenciador de serviços e *Web site* para pesquisas utilizam um núcleo comum de funcionalidades destinadas a ações operacionais e gerenciais provida pela infra-estrutura do gerenciador de serviços (componentes de serviços do gerenciador proposto). Esse núcleo é instalado e configurado no servidor de aplicação para gerenciar os serviços providos pelo próprio ou por outros servidores de aplicação, podendo ser da família J2EE ou .Net. As informações operacionais e gerenciais processadas pelo núcleo são persistidas em uma base de dados (instalada no servidor de bancos de dados).



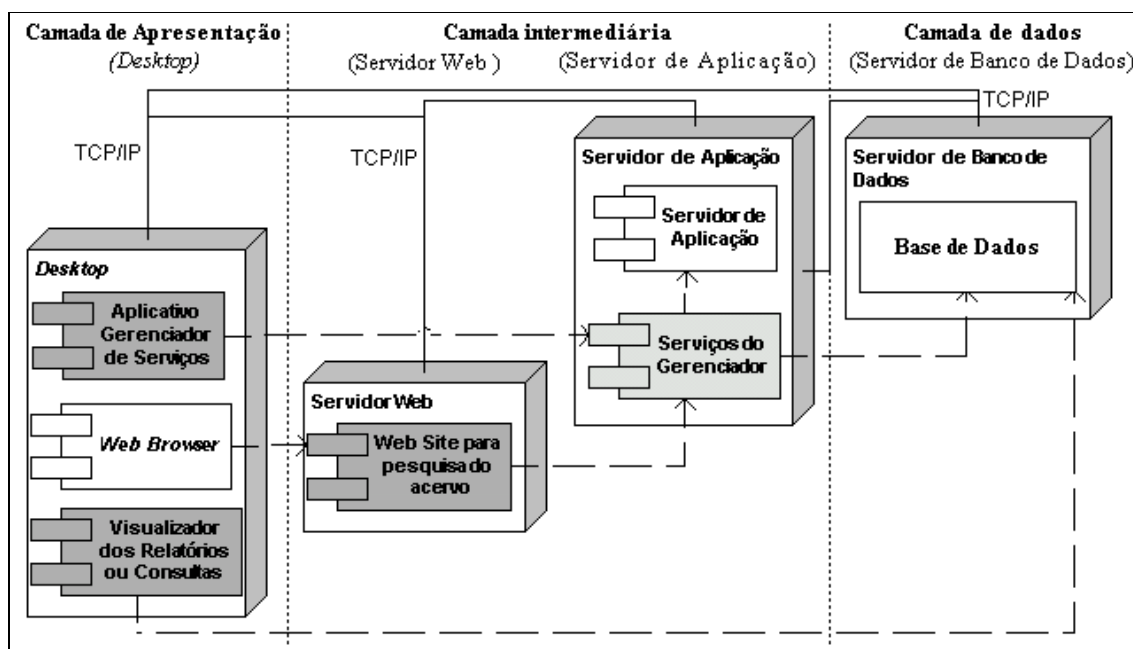


Figura 4.1 – Diagrama de Distribuição e Componentes do Sistema.

### 4.3 Base de dados

Todas as informações contempladas pelo projeto estão especificadas no diagrama de classes da figura 3.4. Entretanto, a persistência dessas informações é realizada pelo paradigma relacional por meio de um sistema gerenciador de banco de dados relacional (SGBDR). A escolha de persistir as informações em banco de dados relacional é direcionada pelos requisitos não funcionais citados no item 3.3 e pelos atuais recursos de *software* disponíveis nas empresas.

Para atender ao requisito de persistência, é aplicada sobre os objetos das classes (instâncias), a técnica de normalização de projetos de banco de dados relacionais (SILBERSCHATZ, KORTH, SUDARSHAN, 1999): evidenciaram-se atributos simples, monovalorados, compostos, multivalorados e complexos. Estes atributos são submetidos a uma engenharia reversa<sup>4</sup> auxiliando a identificação das entidades que compõem o sistema e seus relacionamentos.

Em posse da identificação das entidades e dos seus atributos e relacionamentos é gerado o modelo de dados do projeto, representado na figura 4.2 pelo diagrama entidade-relacionamento do Gerenciador de serviços.

<sup>4</sup> Construção do diagrama E-R a partir das relações na terceira forma normal.

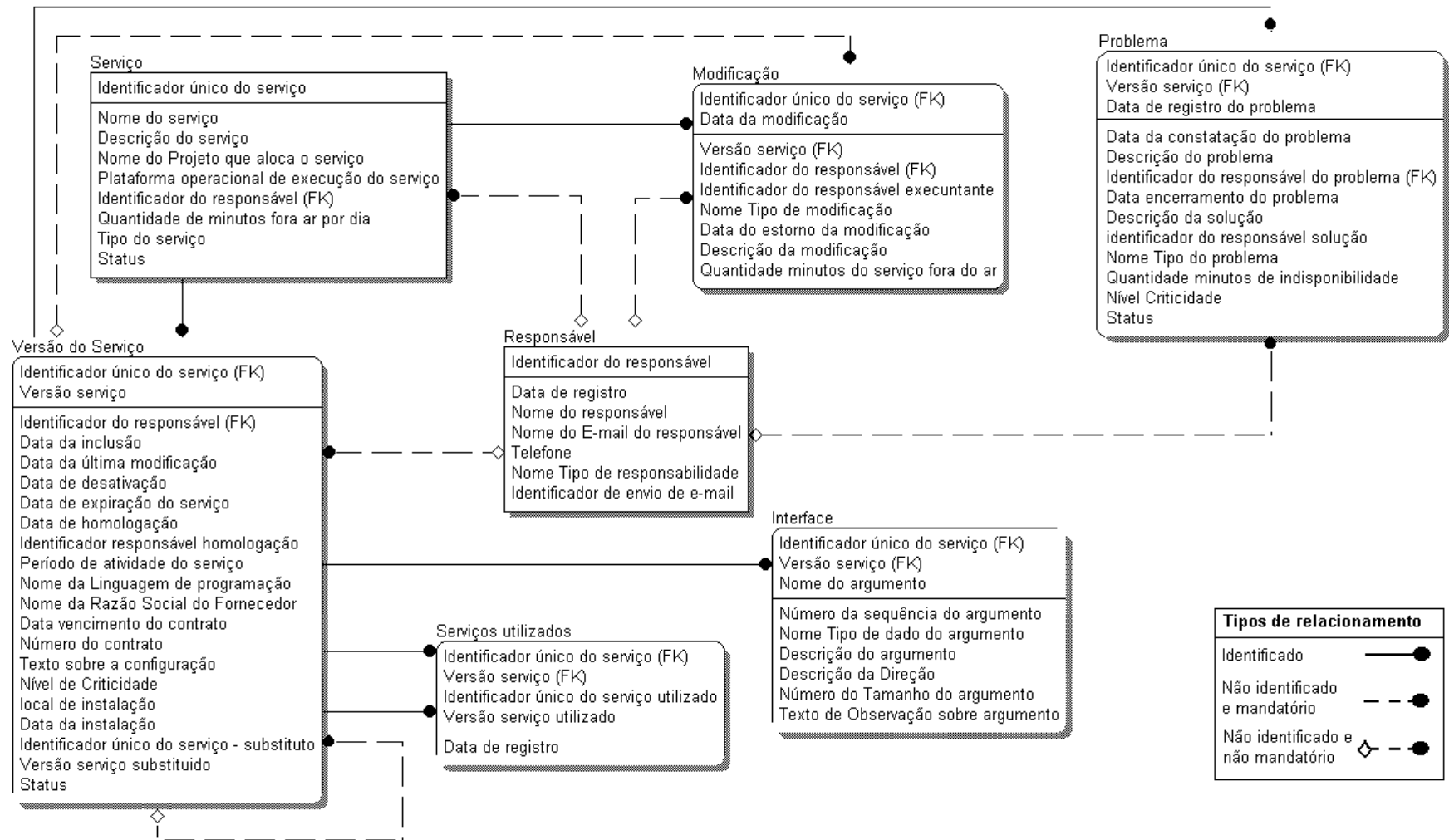


Figura 4.2 – Diagrama Entidade – Relacionamento do Gerenciador de serviços

A figura 4.1 apresenta em um diagrama de distribuição e componentes as partes que compõem o sistema e o seu relacionamento com o SGBDR. A figura destaca os dois consumidores de dados, os serviços usados no gerenciador de serviços e o visualizador dos relatórios ou consultas.

Qualquer método contido no diagrama de classes (Serviços do Gerenciador) realiza as funcionalidades de inserção, alteração e eliminação de dados, por meio de *stored procedures* pertencentes ao SGBDR. As consultas aos dados, utilizadas nos métodos das classes, são feitas por meio de visões, *stored procedures* ou diretamente às tabelas.

Além dos acessos à base de dados efetuados pelos serviços do gerenciador, há os acessos pelo aplicativo visualizador dos relatórios ou consultas. Esse aplicativo externo acessa diretamente as tabelas, as visões ou as *stored procedures* do sistema, para pesquisar diretamente e livremente os dados e estruturá-los de acordo com as necessidades da gestão, gerando informações operacionais e gerenciais (semelhante a uma ferramenta de suporte a decisões - *Decision Support System* – DSS).

#### **4.4 Modelo da Interface Homem-Máquina**

Este item identifica os usuários do sistema, define perfis para os usuários e atribui funções e tarefas a estes perfis. Desse modo, estas ações viabilizam a criação do protótipo das telas do projeto do gerenciador de serviços (Apêndice E).

##### **4.4.1 Usuários do sistema**

O projeto em questão contempla dois grupos distintos de usuários: o primeiro grupo, mais restrito e com diversas atribuições, tais como identificar, implantar, registrar, controlar, administrar e gerir os serviços em um ambiente de produção. O segundo grupo, mais amplo e com poucas atribuições, irá consumir, por meio de pesquisa, as informações desse acervo de serviços, gerar novos serviços, efetuar correções sobre os serviços existentes e auxiliar na descrição de informações.

##### **4.4.2 Perfil dos usuários do sistema**

Para atingir a meta de orientar o projeto para os dois grupos de usuários, há dois tipos de perfis que usam diretamente o sistema: o Administrador dos serviços e o Desenvolvedor dos serviços e aplicativos. Os profissionais com esses dois perfis interagem entre si, durante diversos momentos da vida útil dos serviços, pois são responsáveis pela identificação, criação, instalação, configuração, homologação, manutenção, utilização, eliminação e administração dos mesmos.

De acordo com as informações do item 3.4.1 (atores e casos de uso) esses perfis têm as seguintes atribuições:

- Administrador dos serviços, responsável que detém as seguintes atribuições: instalação dos serviços em ambiente de produção, re-implantação dos serviços decorrentes de manutenções (evolutivas, preventivas, corretivas), desativação e remoção de serviços, gestão das especificações, das modificações, da configuração e versionamentos dos

serviços, dos problemas e soluções que ocorrem com os serviços, dos contratos relacionados aos serviços e divulgação das mudanças que ocorrem com os serviços.

- Desenvolvedor de aplicativos ou serviços, responsável por pesquisar os serviços necessários dentre aqueles disponíveis no acervo, por exemplo, para serem utilizados em novos projetos ou para revisar os serviços que estão sendo usados em projetos ativos. O desenvolvedor também é responsável por fornecer novos serviços para o gerenciador, atualizar serviços existentes, além de homologá-los.

#### 4.4.3 Manual de estilo para o protótipo das telas

O manual de estilo define e padroniza os objetos visuais que participam das telas que compõem o projeto, gerando uma uniformidade gráfica, funcional e de navegabilidade entre os campos da tela, tornando a interface homogênea para os usuários. As definições contidas nesse item orientam a construção do aplicativo cliente GUI e a interface *Web*.

As telas utilizam objetos visuais simples como: janelas de agrupamento, textos, caixas de seleção, caixas de texto, planilhas e botões, entre outros (figura 4.3).

Protótipo de uma interface de usuário. No topo, há um campo de texto com o rótulo "Pesquisar e Selecionar o serviço". Abaixo dele, há um campo de seleção rotulado "Serviço:" e um campo de texto rotulado "Versão:". À direita, há uma tabela com três colunas: "#", "Serviço" e "Versão". Abaixo da tabela, há um botão rotulado "Salvar".

Figura 4.3 – Objetos visuais que compõem as telas.

A orientação visual dos objetos na tela segue priorização na direção horizontal da esquerda para a direita e na direção vertical de cima para baixo. O mesmo ocorre na navegação entre os campos que compõem a tela (figura 4.4).

Protótipo de uma interface de usuário em uma janela intitulada "Gerenciador de Serviços - Cadastro da Interface do Serviço". No topo, há um campo de seleção rotulado "Selecione:" e um campo de texto rotulado "Versão:". Abaixo, há uma tabela com as seguintes colunas: "#", "Nome agrupamento", "Descrição", "Tipo", "Tamanho", "Direção" e "Observação". Na base da janela, há três botões rotulados "Salvar", "Alterar" e "Excluir". Uma seta magenta indica a direção da navegação entre os campos, seguindo a priorização horizontal e vertical.

Figura 4.4 – Direção e sentido da navegação entre os objetos da tela.

As cores dos elementos que compõem as telas, basicamente se concentram nas cores: branca, cinza e em duas intensidades da cor preta.

Os textos da tela que identificam o nome dos atributos dos campos, em geral, são apresentados em negrito. Entretanto, os valores não têm este destaque (figura 4.5).

Descrição: Serviço de autenticação

Figura 4.5 – Forma de apresentação dos campos texto e das caixas de texto.

Os objetos da tela que executam ações de PESQUISA geralmente estão localizados na primeira linha da tela (figura 4.6).

Data término:

Figura 4.6 – Elementos gráficos que compõem a parte inicial da tela.

Os objetos da tela que executam ações de persistência ou eliminação dos dados (SALVAR, ALTERAR ou EXCLUIR) estão localizados na última linha da tela (figura 4.7).

Figura 4.7 – Elementos gráficos que compõem a parte final da tela.

#### 4.4.4 Protótipo das telas

De posse das informações necessárias (requisitos funcionais do projeto, Diagramas de Caso de Uso, Diagrama de Classes, Diagramas de Seqüência, manual de estilo e os requisitos de usabilidade focados nos usuários, perfis e em suas atribuições) são construídas as telas da interface gráfica do sistema que atendem às definições do projeto e à eficiência do seu uso, considerando as seguintes questões citadas por Jakob Nielsen (2000): facilidade de aprendizado, eficiência de uso, facilidade de memorização e baixa taxa de erros (FERNANDEZ, 2005; PÁDUA, 2005). As telas de interface estão desenvolvidas no apêndice E – Protótipo das telas do gerenciador de serviços (figuras E.1 até E.8). Assim, conforme ocorrer a construção efetiva do projeto e, conseqüentemente, os retornos dos usuários (administradores ou desenvolvedores) sobre os testes formais de usabilidade, podem ocorrer modificações nas interfaces.

#### 4.4.5 Procedimentos operacionais

A implantação e o uso no ambiente de produção de uma nova ferramenta como o gerenciador de serviços implica na elaboração de novos procedimentos operacionais, a serem utilizados pelos usuários, para as ocorrências de implantação, re-implantação e desativação dos serviços e para as situações de problemas e de alteração de estado dos mesmos.

No caso do uso do gerenciador de serviços para atuar na implantação de novos serviços, o processo se inicia com uma solicitação de recursos para o administrador emitida pelo desenvolvedor. Nessa solicitação constam as informações que compõem a maior parte das especificações do cadastro do serviço, por exemplo, o nome do serviço, o responsável por seu desenvolvimento, a especificação da interface, os componentes utilizados, entre outras.

Após a análise e a aprovação da solicitação pelo administrador é iniciado o processo de instalação. Durante o processo de aprovação o administrador elege o servidor de aplicação que conterá o serviço. O administrador recebe o novo serviço e o armazena em um local físico padrão no servidor eleito. O administrador ativa o gerenciador de serviços e, com base nas especificações existentes na solicitação, cadastra o novo serviço, a sua interface e os componentes utilizados pelo mesmo (Apêndice E – item E3). No final do processo de cadastramento e ativação do novo serviço, o gerenciador de serviços submete o registro ao servidor de aplicação eleito.

Para as re-implantações de serviços o processo é iniciado com uma solicitação emitida pelo desenvolvedor descrevendo o serviço afetado, o motivo da modificação e a data e horário previsto para a modificação. Após a análise e a aprovação da solicitação pelo administrador, é iniciado, na data e horário previsto, o processo de instalação (a data e hora podem ser negociadas e alteradas). O administrador recebe o serviço modificado e o armazena em um local físico padrão no servidor. O administrador ativa o gerenciador de serviços e, com base nas especificações existentes na solicitação, modifica o cadastro do serviço e, se necessário, altera a interface e os componentes utilizados pelo mesmo (Apêndice E – item E3). No final do processo, o gerenciador de serviços submete o registro do serviço para o servidor de aplicação.

Para a desativação permanente do serviço o processo é iniciado com uma solicitação emitida pelo desenvolvedor descrevendo o serviço afetado, o motivo da desativação e a data e horário previsto para a eliminação. Após a análise e a aprovação da solicitação pelo administrador é iniciado, na data e horário previsto, o processo de desativação (a data e hora podem ser negociadas e alteradas). O administrador executa o gerenciador de serviços e, com base nas especificações existentes na solicitação, elimina o serviço indicado de forma permanente no cadastro de serviço e no servidor de aplicação (Apêndice E – item E3).

O registro de problemas e suas soluções seguem o seguinte procedimento operacional. Geralmente, o administrador é informado pelos usuários que está ocorrendo algum tipo de problema com um determinado aplicativo, o qual, conseqüentemente, utiliza um serviço. O administrador, a partir da suspeita de problemas, inicia uma investigação. Confirmada a existência do problema, o administrador ativa o gerenciador de serviços para registra-la (Apêndice E – item E6). O sistema envia uma mensagem eletrônica de aviso do problema para o desenvolvedor ou responsável pelo serviço. Uma vez resolvido o problema, o administrador aciona o gerenciador de serviços para informar a solução.

Para alterar o estado do serviço o administrador em posse do motivo da alteração e do novo estado, ativa a transação “Altera Estado do Serviço” (Apêndice E – item E8). Uma vez preenchidos os campos obrigatórios desta transação e pressionado o botão “Salvar”, o sistema submete o novo estado para o servidor de aplicação, enquanto salva a atualização no gerenciador de serviços.

#### **4.5 Plano de Testes**

O plano de testes tratado pelo trabalho é dividido em dois itens: descrição e elaboração.

#### 4.5.1 Descrição do plano de testes

O plano de testes é elaborado com base nas especificações funcionais e nos casos de uso do Sistema Gerenciador de Serviços. Os testes visam verificar se o sistema atende os requisitos estabelecidos.

Esse plano de teste é aplicado sobre o gerenciador de serviços cobrindo as funcionalidades de registro, controle e gestão dos serviços.

Os seguintes documentos são utilizados como fontes de informações para o plano de testes.

- Requisitos funcionais do sistema (item 3.2).
- Cenário de Uso do sistema (item 3.4, apêndice A).
- Diagrama de Classes do sistema (item 3.7, figura 3.4 e 3.5).
- Diagrama de Seqüência do sistema (item 3.7, apêndice D).

#### 4.5.2 Elaboração do plano de testes

A etapa de elaboração do plano de testes consiste em selecionar os métodos das classes, especificar os selecionados em detalhes (Apêndice F) e criar sobre essas especificações os casos de teste (Apêndice G).

##### 4.5.2.1 Seleção e especificação dos métodos essenciais

Dentre os diversos métodos de cada classe que compõem o projeto (visualizados nos Diagramas de Classes da figura 3.4 e 3.5, e detalhados no apêndice C), destacam-se para especificação somente os métodos que são essenciais para as funcionalidades de registro, de controle e de gestão dos serviços, propostos para atender as necessidades do projeto do gerenciador de serviços. A síntese dos métodos eleitos está na tabela 4.1 e as especificações detalhadas desses métodos estão no apêndice F - Projeto dos Testes de Integração - Contratos de Classes (BHOR, 2001).

Tabela 4.1 – Lista dos principais métodos.

| Classe      | Métodos                        | Descrição das funcionalidades  |
|-------------|--------------------------------|--|
| Serviço     | Registrar dados do serviço()   | Inclui ou altera dados do serviço contemplando os seus diversos versionamentos. Também permite incluir ou alterar dados dos componentes utilizados pelo serviço. |
|             | Verificar serviços vencidos()  | Consulta data de expiração vencida do serviço.   |
|             | Verificar serviços a vencer()  | Consulta data expiração a vencer do serviço.   |
| Interface   | Registrar dados da interface() | Inclui ou altera dados da interface do serviço.  |
| Modificação | Registrar modificação()        | Inclui nova modificação que ocorreu com o serviço.   |

Continuação da tabela 4.1

|                       |   |  |
|-----------------------|---|--|
| Problema              | Registrar dados do problema()           | Inclui ou altera dados dos problemas que ocorrem com o serviço. O registro de uma solução de um problema consiste em uma simples alteração de dados. |
|                       | Verificar problemas não concluídos()    | Consulta os problemas dos serviços que aguardam solução.   |
| Responsável           | Registrar dados do responsável()        | Inclui ou altera dados dos usuários responsáveis (administradores ou desenvolvedores).   |
|                       | Avisar responsável()                    | Envia e-mail (aviso) para o responsável sobre os aspectos, interfaces e componentes utilizados pelos serviços.                                       |
| Servidor de aplicação | Atualizar dados no servidor aplicação() | Executa as operações de gerência sobre o servidor de aplicação.  |

#### 4.5.2.2 Especificação dos casos de teste

As especificações dos contratos de classe apresentados no apêndice F provêm as informações necessárias para os desenvolvimentos das especificações dos casos de teste que estão apresentadas em detalhes no apêndice G (itens G.1 até G.11).

As especificações dos casos de teste utilizam como modelo em seu desenvolvimento o exemplo de um serviço simples, mas global, de autenticação de usuário que consiste em capturar as informações de usuário e senha (tela de *login* do aplicativo) e, por meio de uma lógica de negócio (serviço de autenticação), verificar se esses dados de entrada (informados pelo usuário do aplicativo) são válidos. A lógica interna de validação do serviço de autenticação pode utilizar as informações contidas em um servidor de banco de dados ou de um controlador de domínio (figura 4.8). Assim, dependendo da necessidade de uso e da generalidade, um serviço de autenticação pode ser utilizado de maneira simultânea por diversos aplicativos.

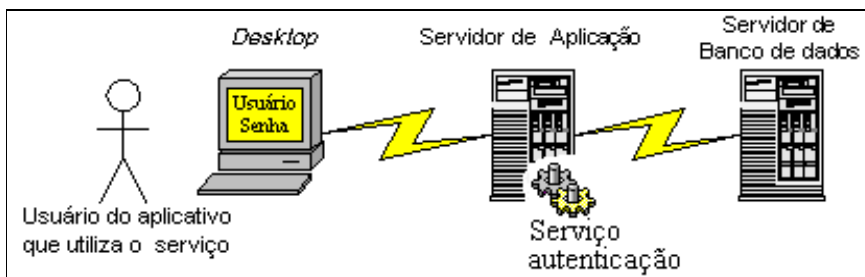


Figura 4.8 – Serviço de autenticação.

Portanto, estes casos de teste sobre o serviço de autenticação de usuário apresentam de forma prática alguns cenários que destacam as funcionalidades de registro, controle e gestão dos serviços e dos respectivos usuários responsáveis.

#### 4.6 Codificação, testes e implantação

A fase de codificação e implantação segue a arquitetura distribuída do projeto (figura 4.1 – Diagrama de distribuição e componentes do sistema) na construção do banco de dados,



dos serviços do gerenciador e dos aplicativos *Web* e GUI formados pelo *Web site* de pesquisas do acervo e pelo aplicativo cliente de gestão.

- Banco de dados do sistema

Com base no diagrama entidade-relacionamento do gerenciador de serviços (figura 4.2) e nas especificações de contrato que utilizam as *stored procedures* (Apêndice F) é iniciada a construção dos objetos que constituem o banco de dados do projeto. Isso incorre na criação de tabelas, de índices, de relacionamento entre os objetos do banco de dados, de visões, de *stored procedures*, e de contas dos usuários que acessam o banco de dados.

- Serviços do gerenciador

Com base nos diagramas de classes (figura 3.4 e 3.5), nos detalhes dos atributos (Apêndice B) e métodos (Apêndice C) das classes, no diagrama de sequência (Apêndice D) e nas especificações de contrato (Apêndice F), é iniciada a construção (codificação) dos serviços do gerenciador fundamentada na programação orientada a objetos, por meio da linguagem *Microsoft Visual Basic*. Os serviços do gerenciador que utilizam o banco de dados relacional realizam as suas tarefas por meio de visões ou *stored procedures* do banco. Após a sua construção, os serviços do gerenciador são instalados em um servidor de aplicação.

- Gerenciador de serviços - Aplicativos GUI e *Web*

- Com base no diagrama de casos de uso (Apêndice A), no diagrama de classes (figura 3.4 e 3.5), no diagrama de sequência (Apêndice D) e no protótipo de telas (Apêndice E), é iniciada a construção (codificação) do aplicativo cliente e do *Web site* que utilizam os serviços do gerenciador. O aplicativo cliente, tanto quanto o *site*, são desenvolvidos em linguagem orientada a objetos por meio da linguagem *Microsoft Visual Basic*. A implantação do aplicativo cliente é realizada sobre as máquinas *desktops*, enquanto que o *site* é instalado em um servidor *Web*, mas acessado pelo cliente por um *Web browser*.
- O aplicativo visualizador dos relatórios ou consultas é tratado na fase de codificação ou implantação, pois apesar de ser um *software* adquirido de terceiros, precisa ser instalado e configurado nas máquinas *desktops*. Para funcionar, o *software* depende da criação de uma conta de usuário no banco de dados que estabeleça acesso somente de leitura nas tabelas, visões ou *stored procedures* que pertencem ao banco de dados (figura 4.1).

Paralelamente à construção do projeto ocorre a fase de testes, que compreende a aplicação das especificações apresentadas no apêndice G para verificar se a construção atende aos requisitos estabelecidos. Testes formais de usabilidade são aplicados pelos administradores e desenvolvedores do sistema.

#### 4.7 Extensibilidade do sistema

A adoção da arquitetura baseada em componentes para o desenvolvimento do sistema gerenciador de serviços possibilita a extensibilidade do mesmo e sua reutilização. Devido a

isso, na evolução do próprio sistema, a mesma modularidade e extensibilidade advinda da componentização protege o investimento do desenvolvimento perante as futuras alterações. O que possibilita que o sistema receba novos requisitos funcionais ou não funcionais, necessitando de um mínimo de interação humana para que estes venham a se integrar.

Desta forma, é possível prover extensões nas especificações das classes dos objetos, por exemplo, para o acréscimo de novos atributos às classes existentes com o objetivo de persistir informações a respeito do desempenho dos serviços ou do ambiente.

Ainda na linha das extensões, é possível construir novas classes de objetos que se referenciem às classes já existentes do sistema, por exemplo, para persistir o nome, o identificador e a versão das aplicações clientes que utilizam um determinado serviço.

#### **4.8 Aderência do trabalho aos objetivos propostos.**

Os requisitos e a maneira como são atendidos no desenvolvimento do trabalho são tratados e destacados neste tópico em quatro itens.

- a) Construção de um ambiente interativo de registro, controle e gestão dos serviços, contemplando o acervo, as configurações e as versões dos serviços, além do registro e do acompanhamento de problemas e soluções que ocorrem com os serviços.

Esses requisitos funcionais são a essência do projeto, portanto são considerados e tratados durante todo o ciclo de desenvolvimento do sistema, desde a análise inicial até a construção do protótipo do projeto. Finalmente, sobre o protótipo, esses requisitos são validados através dos testes funcionais.

Os requisitos funcionais, indiretamente, estão presentes nas seguintes etapas do trabalho: no levantamento das deficiências de gestão dos servidores de aplicação, nos levantamentos e análises dos requisitos, nos diagramas de caso de uso, nos diagramas de classes, nos diagramas de seqüência, no diagrama entidade-relacionamento do gerenciador de serviços, nas especificações de contrato das classes que compõem o projeto, nas telas do protótipo e nas especificações de teste.

- b) Construção de um mecanismo de divulgação automática do acervo e das modificações que ocorrem com os serviços.

Um processo de comunicação *online* que informa os usuários responsáveis o que ocorre com os serviços é fundamental para evitar retrabalhos, conflitos e erros. Também, diminui os gastos de uma empresa no processo de desenvolvimento de serviços ou aplicativos, porque estimula o reuso dos serviços existentes. Desta maneira, o sistema integra o processo de gravação das informações dos serviços, da interface e dos componentes utilizados (Apêndice A, tópico 3, item 1) com o mecanismo que divulga, através de e-mail, o acervo e as modificações dos serviços.

- c) Elaboração de novos procedimentos operacionais para a implantação, para a re-implantação e para a desativação dos serviços, além dos processos de registro dos problemas e das posteriores soluções que ocorrem com os serviços.

A implantação de uma nova ferramenta como o gerenciador de serviços, naturalmente altera a forma de trabalho dos usuários (Administradores e Desenvolvedores), pois automatiza, normaliza e padroniza processos que anteriormente eram manuais, informais e atemporais; viabiliza o registro e controle das operações, e provê novas funcionalidades para a gestão. Desta maneira, para atender às funcionalidades do sistema, foram criados novos procedimentos operacionais com base nas interfaces gráficas do mesmo. Esses procedimentos são descritos no tópico: 4.4 - Interface Homem-Máquina por meio do item 4.4.5 – Procedimentos operacionais.

- d) Construção de um repositório de informações para o sistema, e a liberação de seu acesso para extração de consultas ou relatórios operacionais e gerenciais.

O gerenciador de serviços no cumprimento de suas atividades (requisitos funcionais), utiliza-se de um banco de dados que sustenta as suas operações e funcionalidades. Esse banco de dados persiste e mantém as informações correntes e históricas (registro dos serviços, das modificações e dos problemas). O projeto estende o acesso de seus dados para aplicativos externos, como os geradores e visualizadores de relatórios ou consultas (tópico 4.3 – Base de dados). Para atender a esse consumo de dados dos visualizadores, é criado um acesso de leitura, direto aos objetos do banco de dados (tabelas, visões e *stored procedures*), além da divulgação do diagrama E-R (figura 4.2).

## 4.9 Conclusão

O projeto é composto por várias partes e sua integração é verificada por meio das informações do diagrama de distribuição e componentes (figura 4.1), que mostra todos os nós de rede que participam do projeto e o que deve ser instalado de *software* em cada equipamento de *hardware* atribuído aos nós. Os diagramas de seqüência também auxiliam na verificação da integração das partes, pois implicitamente, nos desenhos destes diagramas, está o fluxo de controle dos processos. Desta forma, estas duas abordagens permitem visualizar, identificar e validar a integração das partes que constituem o sistema.

## CAPÍTULO 5

### 5 CONCLUSÃO

#### 5.1 Resumo

Atualmente os negócios das empresas estão diretamente dependentes de sistemas informatizados, que em sua grande maioria são modularizados e componentizados, ou seja, utilizam na sua composição elementos de pequena granularidade (“tijolos”), os serviços de negócio. A granularidade desses serviços proporciona vantagens, por exemplo, viabiliza o agrupamento de vários desses processos para formar transações mais complexas, facilita a portabilidade desses recursos, e permite o reuso dos recursos em outras aplicações ou sistemas.

Os serviços respondem a duas classes de aplicativos: cliente e servidor. Esses aplicativos, em sua maioria, são executados durante o horário comercial das empresas. Contudo, atualmente, no meio empresarial ocorre um processo crescente de externalização de alguns processos de negócio, principalmente com o uso da Internet (o que indica um processo de globalização das empresas). Assim, os aplicativos e os seus respectivos serviços estão atuando em regime intenso de 24 horas por 7 dias por semana. Essa criticidade dificulta suas manutenções, exigindo assim, uma documentação precisa e muito controle dos serviços e do ambiente.

Os serviços de negócio podem ser construídos com ferramentas distintas que seguem as linhas tecnológicas J2EE ou .NET. Esses serviços são providos por ambientes distintos, proprietários, robustos e especializados, os servidores de aplicação que usam as mesmas tecnologias. Os servidores de aplicação são fornecidos por grandes empresas comerciais e por alguns grupos de *softwares* abertos. Entretanto, são produtos que apresentam, sob alguns aspectos, carências nas funcionalidades de gestão.

Os desenvolvedores de aplicativos ou dos próprios serviços precisam de informações atuais sobre o acervo desses serviços (inventário) para usá-los em novos projetos ou nas manutenções dos existentes, pois podem minimizar o custo dos projetos e o tempo de seu desenvolvimento. Portanto, os desenvolvedores precisam recuperar do inventário as informações a respeito das especificações, das modificações, dos problemas e de suas soluções.

Os gestores e os desenvolvedores devem se preocupar com as implantações ou pós-implantações de novas versões de serviços nesses servidores, motivadas por modificações corretivas, preventivas ou evolutivas, porque podem incorrer em problemas, o que gera indisponibilidade que, conseqüentemente, acarreta perdas financeiras para a empresa.

Esse cenário amplo e complexo que envolve os desenvolvedores e gestores, os serviços e os seus vários ambientes provedores são tratados pelo trabalho por meio da proposta do projeto de um gerenciador de serviços. Esse gerenciador estende as funcionalidades da gestão nativa desses produtos (servidores de aplicação) permitindo documentar, catalogar, controlar e gerenciar os serviços providos pelos vários servidores, além de prover para os desenvolvedores recursos de pesquisa sobre as informações do acervo.

## 5.2 Análise de resultados e comentários

De maneira geral, as idéias e a proposta do trabalho estão em conformidade com a realidade, pois já existem fornecedores incluindo em seus produtos (servidores de aplicação) a gestão de versões (uma das funcionalidades propostas), o que era muito raro presenciar no início das pesquisas do trabalho em 2004. Entretanto, o retorno dos benefícios dessas inclusões funcionais nos produtos é limitado, devido ao fato de serem soluções proprietárias. Esses produtos possuem limites nas informações que são registradas o que restringe, conseqüentemente, o controle e a gestão. Por exemplo, não registram as especificações de contrato, de homologação, dos componentes utilizados, das modificações ocorridas, dos problemas, e outras informações. Os produtos não disponibilizam acesso à base de dados das suas informações para as pesquisas dos desenvolvedores e para a criação de análises operacionais e gerenciais dos administradores. Essas limitações tornam-se mais acentuadas em empresas que utilizam diferentes produtos proprietários (por exemplo, os servidores de aplicação BEA, IBM, JBoss e .NET) que exigem diferentes formas de registro, controle e gestão, ou seja, há falta de uniformidade e padronização nestes processos, devido a necessidade de independência dos fornecedores motivados pela concorrência.

A proposta do projeto estende o gerenciamento dos serviços, o que torna possível documentar (registrar), categorizar, organizar, padronizar, operacionalizar e administrar a maioria dos serviços (desenvolvidos ou adquiridos) que existem em uma empresa, ou seja, trata o inventário de serviços de uma forma única, padronizada e independente do servidor de aplicação ou dos servidores de aplicação eleitos, mas estabelece com eles uma comunicação operacional e gerencial (API). Os ganhos dessa independência na gestão, ofertados por um sistema único e especializado para esta finalidade, o gerenciador de serviços, ocorrem à medida que nele são adicionados e tratados os diversos serviços dos outros servidores. Esses recursos trazem vantagens de registro, de controle e de gestão, além de disponibilizar para pesquisas e análises a extração de dados de uma base única.

O inventário é a base para as divulgações e pesquisas que estimulam o reuso dos serviços, pois contém o catálogo dos mesmos. Desta maneira, o projeto permite além das informações que identificam cada serviço que compõe o inventário, registrar e gerir as especificações referentes a datas de controle, a homologação, a interface, o controle dos contratos, a utilização, o ambiente, as configurações e versões, os usuários responsáveis, os componentes utilizados pelo serviço, as alterações de estado, as modificações e os problemas e suas respectivas soluções.

Sendo assim, o projeto cumpriu o escopo do trabalho apresentado no tópico 1.2 – Objetivo, pois gerou um novo ferramental de gestão dos serviços, o qual depende de vários detalhes informativos e, assim, para a administração dos serviços utiliza novos procedimentos para implantar, re-implantar, desativar, ativar, desativar permanentemente e registrar as modificações, os problemas e suas soluções. O resultado do uso deste ferramental acrescenta mais informações para a própria gestão, para as consultas e relatórios operacionais e gerenciais e também para o *site* de pesquisa. Além disso, o projeto atual possibilita extensibilidade funcional, o que torna possível a adição de novas funcionalidades que surgem naturalmente com maturidade dos processos e com o uso da ferramenta.

À medida que ocorreu o desenvolvimento do trabalho foi possível identificar no projeto algumas restrições, tais como:

- O gerenciador de serviços não é um aplicativo *Web*, tal qual o site para pesquisas, pois a princípio a idéia é ter em um número reduzido, identificado e controlado de estações de gerenciamento para esse aplicativo de administração.
- A gestão de um serviço através do gerenciador de serviços é para um único e determinado servidor de aplicação, pois o projeto atual somente controla um local de instalação por serviço.

Durante o desenvolvimento do projeto surgiram algumas questões e problemas que embora solucionados, são pertinentes de serem citados.

- Dificuldade em obter e selecionar informações a respeito do gerenciamento de serviços. Durante as pesquisas sobre o gerenciamento de serviços não foram encontradas muitas informações. Sendo assim, a solução foi explorar a questão partindo da definição global de gerenciamento e adicionando a ela, informações relevantes sobre a gestão dos *Web Services*, gerência de configuração, versões e das modificações de *software* e o reuso de componentes.
- Dificuldade em obter e selecionar informações a respeito da administração e do gerenciamento providos pelos servidores de aplicação. Durante a fase de levantamento e análise de informações dos servidores de aplicação ocorreu grande dificuldade na identificação de informações sobre a administração dos serviços, porque o foco principal das documentações desses produtos está no potencial de provimento de serviços com segurança, integração de sistemas e alta disponibilidade com alta eficiência (item 2.3). Entretanto, mais pesquisas em diversos documentos, conversas com os especialistas de alguns dos produtos (IBM, BEA e BORLAND) e testes viabilizaram o levantamento das informações sobre esta administração.
- Dificuldade para compatibilizar a persistência dos objetos OO originária dos serviços do gerenciador, com o banco de dados relacional. Porque, no modelo OO as classes representam logicamente os objetos alvo do sistema (Serviços, Interface, entre outros), através de atributos próprios e comportamentos definidos, enquanto que o modelo relacional de um sistema OLTP (*On-Line Transaction Processing*) prioriza a normalização dos dados. A solução é aplicar uma análise relacional de normalização sobre os objetos das classes (instâncias) do sistema para construir a partir desta o diagrama ER (figura 4.2).

A implantação do protótipo foi feita sobre a plataforma Windows por causa de seu uso intensivo em ambiente empresarial, e foi adotado um SGBDR visto que alguns tipos destes produtos já existem na maioria das empresas, o que minimiza os custos diretos ou indiretos. Entretanto, o projeto pode ser facilmente implementado de forma parcial ou completa sobre outras plataformas, pois utiliza uma arquitetura distribuída que prioriza a dedicação e a especialização dos ambientes que a compõem (SGBDR, provedor de serviços, ambiente *desktop* para o aplicativo cliente), isto facilita a mobilidade, a disponibilidade, a aplicação de critérios de segurança, o desempenho e a evolução desses ambientes e de seus respectivos elementos. Além da arquitetura, a mobilidade ou portabilidade para outras plataformas é favorecida nesse projeto pela tecnologia de orientação a objetos que independe da plataforma,

ela especifica os objetos que compõem o sistema através do diagrama de classes, os distribui sobre a rede de comunicação através do diagrama de distribuição e componentes, e possibilita para os desenvolvedores a escolha entre várias ferramentas de programação que atendem a diversas plataformas (*Visual Basic* para .NET, *WebSphere* para J2EE, entre outras).

### 5.3 Contribuições

A principal contribuição desse trabalho é a forma abstrata e independente do registro, controle e gestão dos serviços perante a variedade de servidores de aplicação. Essa contribuição é viabilizada por meio da construção de um sistema de gestão dos serviços provido por diversos servidores de aplicação, que os trata de forma uniforme, padronizada e organizada. Essa gestão acompanha o ciclo de vida dos serviços no ambiente de produção, iniciando com a instalação e configuração dos serviços, seguindo com as re-instalações e re-configurações dos mesmos (motivadas por modificações: evolutivas, preventivas e corretivas), e terminando com suas desativações.

Durante o processo de gestão, o gerenciador de serviços, armazena em repositório único (inventário) as informações do acervo, as especificações de interface, o registro dos componentes utilizados, as modificações, os problemas e as soluções que ocorreram e ocorre com os serviços.

Além da gestão dos serviços, a proposta do trabalho contribui para a construção de um ambiente propício para reuso dos serviços (PRESSMAN, 2001), pois fornece uma ferramenta para a catalogação e pesquisa dos serviços (item 2.2 - biblioteca eletrônica) que, geralmente, estão espalhados no parque de *softwares* das empresas.

Dessa forma, o esforço dedicado à elaboração do trabalho culmina no resultado do desenvolvimento do projeto, o protótipo funcional do Gerenciador de serviços, que é capaz de viabilizar as funcionalidades de registro, pesquisa, controle e gestão dos serviços, conforme o que estava estipulado no objetivo e na especificação dos casos de teste (apêndice G).

A seguir estão citadas as contribuições indiretas proporcionadas pelo trabalho.

- Identificação do gerenciador de serviços como uma necessidade do mercado de *softwares*.
- Utilização de boas práticas da engenharia de *software* no desenvolvimento do projeto, viabilizando assim, através do documento final um estudo de caso que poderá instruir o desenvolvimento de trabalhos similares. Entre as boas práticas estão:
  - Segmentação e elaboração do desenvolvimento do projeto nas etapas clássicas do ciclo de desenvolvimento cascata;
  - Utilização da análise orientada a objetos (OO);
  - Formalização da análise OO através dos diagramas da UML, tais como, diagrama de caso de uso, diagrama de sequência, diagrama de classes e o diagrama de distribuição e componentes;

- Atendimento a requisitos de usabilidade na criação do manual de estilo e nas telas de interface;
- Especificações dos contratos de classe em prol dos casos de testes;
- Especificação e elaboração de um banco de dados relacional.

#### 5.4. Sugestão para futuros trabalhos

Durante o decorrer do desenvolvimento do trabalho foram surgindo novas idéias, necessidades e tecnologias que possibilitaram a criação de futuras versões do gerenciador de serviços, que contemplem os seguintes itens:

- Acrescentar ao gerenciador de serviços, o registro, o controle e o gerenciamento dos consumidores (solicitantes) de serviço, os aplicativos clientes. Pesquisas sobre estas novas informações ajudam a prever e a diminuir impactos nas modificações (evolutivas, preventivas, corretivas) dos serviços.
- Acrescentar as ações de monitoramento e de alertas sobre desempenho dos serviços e do ambiente. Tratar o resultado destas ações de tal maneira que realmente a gestão, tornando-a mais pró-ativa.
- Acrescentar o registro e o controle nas intervenções de segurança durante o processo de instalação (ou re-instalação) dos serviços sobre o servidor de aplicação. Essas novas informações viabilizam análises de auditoria, além de identificar, controlar e habilitar os respectivos usuários para cada serviço.
- Converter todo o sistema para a *Web* visando uma única plataforma de administração e pesquisa de informações, uma mobilidade maior de uso e uma expansão do público consumidor.
- Integrar o gerenciador de serviços com o ambiente *Web Services* viabilizando extensões na gestão e provendo atualizações através dos padrões UDDI e WSDL. Assim, através de uma interface de gerenciamento, o Gerenciador de serviços pode integrar-se com os *Web Services* (similar a integração do próprio Gerenciador de serviços com os Servidores de aplicação), submetendo a eles operações de gerência, tais como, o registro das especificações dos serviços descritas na linguagem XML de acordo com os padrões exigidos pelo WSDL. Esse mesmo mecanismo de integração permitirá a publicação do acervo, das interfaces e dos provedores dos serviços de acordo com o padrão UDDI.
- Estabelecer, de maneira periódica, um monitoramento funcional dos serviços, por meio da atribuição de valores padrão de teste para os argumentos que compõem o serviço, de tal maneira que gerem também uma resposta padronizada. Seria o equivalente a monitorar os pinos de teste existentes em placas de circuito impresso ou *chips*. Dessa forma, os dados coletados desta revisão periódica tornam-se o *baseline* de comparação de desempenho sobre as próximas re-implantações do mesmo serviço.
- Dividir, isolar e segregar logicamente a gestão do projeto proposto pelo gerenciador de serviços para os ambientes de desenvolvimento e produção dos serviços. Nessa



segregação o uso do gerenciador de serviços será atribuído para o perfil dos desenvolvedores de serviços que registraram as pré-implantações dos serviços (concepção, projeto, primeiras compilações e a versão Beta) e as modificações destes, enquanto que a outra parte atua com a proposta atual. Essa segregação da gestão viabiliza uma integração com o ciclo de desenvolvimento dos serviços e facilita a implantação final (no ambiente de produção) dos serviços concluídos e homologados.

- Desenvolver no gerenciador de serviços um recurso de *disaster recovery* para os serviços e para as configurações dos servidores de aplicação. Esse recurso utiliza as informações contidas na base de conhecimento do gerenciador de serviços para que o processo de *disaster recovery* possa instalar, reinstalar e configurar esses serviços em servidores de aplicação existentes ou novos.

## Referências Bibliográficas

AHOKAS, T. **Using XML in making Legacy application data accessible through web services** – March 31, 2003.

Disponível em: <<http://www.tml.hut.fi/~tpv/opiskelijat/ahokas.pdf>>. Acesso em: 06 set. 2004.

ALMEIDA, E. **Método de Teste Baseado na Especificação de Requisitos do Software para Projetos Orientado a Objetos**. 2001. Dissertação (Mestrado em Engenharia de Computação) – Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, 2001.

AREVOLO, W. **Portals: Gaining Advantage From Web-Focused Integration** – Palestra da Gartner Group – Application Integration and Web Services 2003.

Disponível em: <[http://www3.gartner.com/2\\_events/conferences/apn8/apn8\\_1.jsp](http://www3.gartner.com/2_events/conferences/apn8/apn8_1.jsp)>. Acesso em: 23 fev. 2005.

ARSANJANI, A. **Developing and Integrating ENTERPRISE COMPONENTS AND SERVICES** – Communication of the ACM – October 2002/Vol 45. No.10.

Disponível em: <<http://portal.acm.org/citation.cfm?id=570907.570926>>. Acesso em: 01 mar. 2005.

BARRY & ASSOCIATES **Application server articles** – 2004

Disponível em: <<http://www.service-architecture.com>>. Acesso em: 10 nov. 2004.

BAX, M.; LEAL, G. **Serviços Web e a evolução dos serviços em TI** – DataGramaZero – Revista de Ciência da Informação – v.2 n.2 abril/01

Disponível em: <[http://www.dgz.org.br/abr01/Art\\_02.htm](http://www.dgz.org.br/abr01/Art_02.htm)>. Acesso em: 29 jul. 2004.

BEA WebLogic Server 7.0 **Overview - Administration and Management** – April 2, 2002. Disponível em:

<[http://safarixamples.informit.com/0130463868/pdfs/wl\\_server\\_overview\\_wp.pdf](http://safarixamples.informit.com/0130463868/pdfs/wl_server_overview_wp.pdf)>. Acesso em: 03 mar. 2003.

BHOR, A. **Software Component Testing Strategies** – Technical report UCI-ICS-02-06, June 2001.

Disponível em: <<http://www.ics.uci.edu/...>>. Acesso em: 21 jun. 2005.

BOOTH, D. et al. **Web Services Architecture** – February 2004

Disponível em: <<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>>. Acesso em: 11 nov. 2004.

BORLAND **Borland Enterprise Server AppServer Edition** - 2005.

Disponível em: <<http://www.borland.com/bes/index.html>>. Acesso em: 15 mai. 2005.

BORLAND **VisiBroker** – 2005.

Disponível em: <<http://www.borland.com/visibroker/>>. Acesso em: 15 mai. 2005.

BURBECK, S. **The Tao of e-business services – The evolution of Web applications into service-oriented components with Web services** – Oct 2000

Disponível em: <<http://www-106.ibm.com/developerworks/webservices/library/ws-tao/>>. Acesso em: 13 nov. 2004.

COYLE, F. **XML, Web Services and the Changing Face of Distributed Computing** – Ubiquity An ACM IT Magazine and Forum - 2001.

Disponível em: <[http://www.acm.org/ubiquity/views/f\\_coyle\\_1.html](http://www.acm.org/ubiquity/views/f_coyle_1.html)>. Acesso em: 22 jul. 2003.

CRNKOVIC, I; LARSSON, M. **Component-Based Software Engineering – New Paradigm of Software Development** - 2001

Disponível em: <<http://www.mrtc.mdh.se/publications/0293.pdf>>. Acesso em: 23 fev. 2005.

CYBIS, W. **Gerenciamento de configuração** - Laboratório de Utilizabilidade da Informática - Departamento de Informática e Estatística - Universidade Federal de Santa Catarina – Novembro, 2003.

Disponível em: <[http://www.inf.ufsc.br/~cybis/ine5617/Gerenciamento\\_configuracao.ppt](http://www.inf.ufsc.br/~cybis/ine5617/Gerenciamento_configuracao.ppt)>. Acesso em: 17 nov. 2004.

FEIMAN, J. **Application Server Scenario: .NET, J2EE and Application Platform Suites** – Palestra da Gartner Group – Integração de Aplicativos e Serviços na Web 18-19 Março 2003.

Disponível em: <[www.gartner.com/2\\_events/conferences/2003/apps6i/apps6i.pdf](http://www.gartner.com/2_events/conferences/2003/apps6i/apps6i.pdf)>. Acesso em: 06 out. 2004.

\_\_\_\_\_. **What Will Web Services Do to Application Development?** – Palestra da Gartner Group – Application Integration and Web Services 14-15 April 2004.

Disponível em: <[https://www.g2r.com/2\\_events/conferences/2004/mex16l/mex16l\\_1a.jsp](https://www.g2r.com/2_events/conferences/2004/mex16l/mex16l_1a.jsp)>. Acesso em: 09 set. 2005.

\_\_\_\_\_. **Application Platform Scenario: From Battle of the Servers to Battle of the Suites** – Palestra da Gartner Group – Application Integration and Web Services 14-15 April 2004.

Disponível em: <[http://www3.gartner.com/2\\_events/conferences/2003/apn11/apn11\\_1a.jsp](http://www3.gartner.com/2_events/conferences/2003/apn11/apn11_1a.jsp)>. Acesso em: 01 mar. 2005.

\_\_\_\_\_. **Tapping Real Productivity: The Service-Oriented Development Scenario** – IV Conferência Anual Gartner – Integração de Aplicativos e Serviços na Web – 30-31 de março de 2005.

Disponível em: <[http://www.gartner.com/2\\_events/conferences/2004/apn13/CIODayAgenda.pdf](http://www.gartner.com/2_events/conferences/2004/apn13/CIODayAgenda.pdf)> Acesso em: 07 jun. 2005.

\_\_\_\_\_. **Global Trends in Application Development** - IV Conferência Anual Gartner – Integração de Aplicativos e Serviços na Web – 30-31 de março de 2005.

Disponível em: <[http://gartner11.gartnerweb.com/2\\_events/conferences/brl13l/brl13le\\_1a.jsp](http://gartner11.gartnerweb.com/2_events/conferences/brl13l/brl13le_1a.jsp)>. Acesso em: 09 set. 2005.

FERNANDEZ, A. **Usabilidade: um pouco da história e definição.**

Disponível em: <<http://www.webinsider.com.br/vernoticia.php?id=2398>>. Acesso em: 02 jun. 2005.

FOWLER, M; SCOTT K. **UML Distilled: a brief guide to the standard object modeling** – Second Edition – 2000 – ISBN 0-201-65783-X.

GLASS, G. **The Web services (r)evolution, Part 4 - Web Services Description Language (WSDL)** – Feb 2001.

Disponível em: <<http://www-106.ibm.com/developerworks/webservices/library/ws-peer4/>>.

Acesso em: 13 nov. 2004.

HOCHMULLER, E.; MITTERMEIR, R.; POZEWAUNIG, H. **Software – Entwurf, Test und Entwicklungsprozess - Component Based Software Development** - March, 2002.

Disponível em: <[http://www.isys.uni-](http://www.isys.uni-klu.ac.at/ISYS/Courses/02WS/sete/folien/Fse0209_CBSD.pdf)

[klu.ac.at/ISYS/Courses/02WS/sete/folien/Fse0209\\_CBSD.pdf](http://www.isys.uni-klu.ac.at/ISYS/Courses/02WS/sete/folien/Fse0209_CBSD.pdf)>. Acesso em: 23 fev. 2005.

HOTLE, M. **Web Services Scenario: Poised for the Mainstream** – Palestra da Gartner Group – Application Integration and Web Services 14-15 April 2004

Disponível em: <[http://www4.gartner.com/teleconferences/asset\\_62752\\_75.jsp](http://www4.gartner.com/teleconferences/asset_62752_75.jsp)>. Acesso em:

23 fev. 2005.

\_\_\_\_\_. **The Application Development Scenario** – Palestra da Gartner Group – Application Integration and Web Services 14-15 April 2004

Disponível em: <[https://www.g2r.com/2\\_events/conferences/2004/mex16l/mex16l\\_1a.jsp](https://www.g2r.com/2_events/conferences/2004/mex16l/mex16l_1a.jsp)>.

Acesso em: 09 set. 2005.

IBM. IBM – **Software – Tivoli – Introduzindo o Tivoli Kernel Services Administration** – 2001.

Disponível em:

<[http://publib.boulder.ibm.com/tividd/td/TKS/tks\\_12\\_introducing/pt\\_BR/HTML/adminmstftrm.htm](http://publib.boulder.ibm.com/tividd/td/TKS/tks_12_introducing/pt_BR/HTML/adminmstftrm.htm)>. Acesso em: 03 mar. 2005.

INFORMATIVO TÉCNICO No. 70 (jan/fev/2000) **Modelo CMM de qualidade de software**

Disponível em: <<http://www.revista.unicamp.br/infotec/informacao/inf70.htm>>. Acesso em:

17 nov. 2004.

JBoss Application Server 4.0

Disponível em: <[http://www.jboss.com/pdf/JBoss\\_AS\\_4.0\\_Datasheet.pdf](http://www.jboss.com/pdf/JBoss_AS_4.0_Datasheet.pdf)>. Acesso em: 28 ago. 2004.

KRILL, P. **Sun readies app server upgrade** - INFOWORLD - August 31, 2004

Disponível em: <[http://www.infoworld.com/article/04/08/31/HNsunappserver\\_1.html](http://www.infoworld.com/article/04/08/31/HNsunappserver_1.html)>.

Acesso em: 24 fev. 2005.

LARSSON, M. **Applying configuration management techniques to component-based systems** – December 2000.

Disponível em:

<<http://citeseer.ist.psu.edu/cache/papers/cs/17997/http:zSzzSzwww.mrtc.mdh.sezSzpublicatio nszSz0262.pdf/larsson00applying.pdf>>. Acesso em: 24 fev. 2005.

LHEUREUX, B. **Application Integration Scenario: Impact of the New Technology Is Finally Clear** – Palestra da Gartner Group **Conferência sobre a Integração de Aplicativos e Serviços na Web** – Março de 2003.

Disponível em: <[http://www3.gartner.com/2\\_events/conferences/apn9/apn9.pdf](http://www3.gartner.com/2_events/conferences/apn9/apn9.pdf)>. Acesso em:

24 fev. 2005.

LICHT, D. **Gestão, TI e Gerenciamento de Configuração** – Revista TI – 29 de outubro de 2004.

Disponível em: <[http://www.timaster.com.br/revista/artigos/main\\_artigo.asp?codigo=959](http://www.timaster.com.br/revista/artigos/main_artigo.asp?codigo=959)>. Acesso em: 17 nov. 2004.

LIU, X. *et al.* **Building reliable, high-performance communication systems from components** - 1999. Department of Computer Science. Cornell University, Ithaca, NY 14850. Disponível em: <<http://www.cs.cornell.edu/home/rvr/papers/sosp99/paper.pdf>>. Acesso em: 27 dez. 2002.

MCCOY, D.; FEIMAN, J. **Application Server Scenario: .NET, J2EE and Application Platform Suites** – Palestra da Gartner Group Conferência sobre a Integração de Aplicativos e Serviços na Web – Março de 2003.

Disponível em: <[http://www.gartner.com/2\\_events/conferences/2003/mex14l/mex14l\\_1.jsp](http://www.gartner.com/2_events/conferences/2003/mex14l/mex14l_1.jsp)>. Acesso em: 09 set. 2005.

MOHAN, C. **Panel – Application Servers: Born-Again TP Monitors for the Web?** - 2001

Disponível em: <<http://portal.acm.org/citation.cfm?id=375663.375798>>. Acesso em: 24 fev. 2005.

\_\_\_\_\_. **Tutorial: Application Servers and Associated Technologies** – ACM SIGMOD International Conference on Management of Data (SIGMOD'02), Madison, USA, June 2002. Disponível em:

<[http://www.almaden.ibm.com/u/mohan/AppServersTutorial\\_SIGMOD2002.pdf](http://www.almaden.ibm.com/u/mohan/AppServersTutorial_SIGMOD2002.pdf)>. Acesso em: 24 fev. 2005.

MOUGIN, P.; BARRIOLADE, C. **Web Services, Business Objects and Component Models** – Orchestra Networks – July 2001.

Disponível em: <[http://www.xml.org/xml/feature\\_articles/orchestra\\_webservice.pdf](http://www.xml.org/xml/feature_articles/orchestra_webservice.pdf)>. Acesso em: 10 set. 2003.

NATIS, Y. **Selecting an Application Platform: The Path to the Perfect Software Stack** – IV Conferência Anual Gartner – Integração de Aplicativos e Serviços na Web – 30-31 de março de 2005.

Disponível em: <[http://www.gartner.com/2\\_events/conferences/2005/brl22l/brl22l\\_1.jsp](http://www.gartner.com/2_events/conferences/2005/brl22l/brl22l_1.jsp)>. Acesso em: 07 jun. 2005.

NIELSEN, J. **Projetando Websites**. Editora Campus, 2000. Tradução de "Designing Web Usability: The Practice of Simplicity", da New Riders Publishing, 2000 – ISBN 8535206566.

PÁDUA, C. **Apostila de Engenharia de Usabilidade** – ICEx / UFMG – Departamento da Ciência da Computação – Laboratório Gestus/Synergia – Abril / 2005.

Disponível em: <<http://www2.dcc.ufmg.br/~clarindo/arquivos/disciplinas/eu/material/referencias/apostila-usabilidade.pdf>>. Acesso em: 28 jul. 2005.

PRESSMAN, R. **Software Engineering: A practitioner's approach** – 5<sup>th</sup> ed. - 2001 – ISBN 0-07-365578-3

PURSNANI, V. **An Introduction to Java Servlet Programming** – November 2001.

Disponível em: <<http://www.acm.org/crossroads/xrds8-2/servletsProgramming.html>>. Acesso em: 21 jul. 2003.

SACHET, D.; ARAKAKI M. **Aplicabilidade de CORBA na Implementação de TMN** – Out/2000.

SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S. **Sistema de Banco de Dados** – São Paulo: MAKRON Books, 1999 – ISBN: 85.346.1073-8.

SILVA, F. **Reutilização de software através de geração de código e de desenvolvimento de componentes – Estudo de caso** – 2003.

Disponível em: <<http://www.pcs.poli.usp.br/posgrad/teses/Lteses2003.asp>>. Acesso em: 24 fev. 2005.

SKONNARD, A. **The XML files - The Birth of Web Services**– MSDN magazine – October 2002

Disponível em: <<http://msdn.microsoft.com/msdnmag/issues/02/10/XMLFiles/>>. Acesso em: 24 fev. 2005.

SYSTINET **Introduction to Web Services Architecture** – 2002.

Disponível em:

<[http://www.systinet.com/download/74c83b5cf8607b124e247426574e0943/wp\\_Systinet\\_SO\\_A.pdf](http://www.systinet.com/download/74c83b5cf8607b124e247426574e0943/wp_Systinet_SO_A.pdf)>. Acesso em: 24 fev. 2005.

SYSTINET **Four “Abilities” SOA will lack without a Registry** – July 2004.

Disponível em: <[http://www.systinet.com/download/wp\\_abilities\\_note.pdf](http://www.systinet.com/download/wp_abilities_note.pdf)>. Acesso em: 01 mar. 2005.

TROFIN, M. **A Framework for Removing Redundant Context Management Services in Enterprise JavaBeans Application Servers** – October 2004.

Disponível em:

<<http://kilimanjaro.eas.asu.edu/servlets/Search?index=%3Ctitle%3Ea+framework+for+removing+redundant+context+management+services+in+enterprise+javabeans+application+servers%3Cauthor%3EM.+Trofin%3Cyear%3E2004>>. Acesso em: 11 nov. 2004.

\_\_\_\_\_. **Optimizing Context Management in Contextual Composition Frameworks** – 2004.

Disponível em: <<http://www.eeng.dcu.ie/~trofinm/projects.htm>>. Acesso em: 13 dez. 2004.

WORTHEN, B. **Web Services Still Not Ready for Prime Time** - CIO Magazine - September 1, 2002

Disponível em: <<http://www.cio.com/archive/090102/prime.html>>. Acesso em: 26 out. 2004.

## **Referências consultadas**

AGRAWAL, R. *et al.* **Vinci: A Service-Oriented Architecture for Rapid Development of Web Applications** – 2001 ACM 1-58113-348-0/01/0005.

Disponível em: <<http://portal.acm.org>>. Acesso em: 06 ago. 2003.

Apple Computer, Inc. **MAC OS x Server – Java Application Server Administration for version 10.3 or later** – January 22, 2004.

Disponível em: <[http://images.apple.com/server/pdfs/JavaApplication\\_Server.pdf](http://images.apple.com/server/pdfs/JavaApplication_Server.pdf)>. Acesso em: 10 mar. 2005.

ARULANTHU, A. *et al.* **The Design and Performance of a Scalable ORB Architecture for CORBA Asynchronous Messaging** – Washington University, St. Louis, MO 63130. Disponível em: <<http://www.cs.wustl.edu/~schmidt/PDF/ami2.pdf>>. Acesso em: 14 out. 2003.

BENGTSSON, P. **A Component Implementation Variations Study** – University of Karlskrona/Ronneby - 2000. Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>. Acesso em: 09 fev. 2004.

BLOM, M.; NORDBY, E. **Semantic Integrity in Component Based Development** – 2000. Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>. Acesso em: 09 fev. 2004.

BRITTON, C. **IT Architectures and Middleware: strategies for building large, integrated systems** – 2001 – ISBN: 0-201-70907-4.

CARRUTHERS, D.; DAWKINS, H.; MAMAREDDY, P. **CORBA vs. DCOM** [s.d.] Disponível em: <<http://www.cs.auckland.ac.nz/~john-g/teaching/482A/presentations/presentation8/sld001.htm>>. Acesso em: 14 out. 2003.

CHAMPION, M. *et al.* **W3C - Web Service Architecture** – November 2002. Disponível em: <<http://www.w3.org/TR/2002/WD-ws-arch-20021114>>. Acesso em: 10 nov. 2004.

CHEN, D. **Towards A Software Component Framework for Mechatronics Applications** – Royal Institute of Technology, Sweden – 2000. Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>. Acesso em: 09 fev. 2004.

CHRISTIANSSON, B.; JAKOBSSON, L. **Component-Based Software Development Life Cycles** – Karlstad University – 2000. Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>. Acesso em: 09 fev. 2004.

CRNKOVIC, I. **Component-based Software Engineering: Building Systems form Software Components**. Vasteras, Sweden. Disponível em: <<http://www.mrtc.mdh.se/publications/0405.pdf>>. Acesso em: 18 dez. 2002.

CRNKOVIC, I; LARSSON, M. **A case study: Demands on Component based Development** – Vasteras, Sweden - 2000. Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/ICSE2000-final.pdf>>. Acesso em: 18 dez. 2002.

CRNKOVIC, I; LARSSON, M.; KUNG-KIU, L. **Component Configuration Management for Frameworks**. – 1999.

Disponível em: <[http://www.idt.mdh.se/~icc/articles/1999-apsec/ccminframeworks\\_final.pdf](http://www.idt.mdh.se/~icc/articles/1999-apsec/ccminframeworks_final.pdf)>. Acesso em: 02 jan. 2003.

CRNKOVIC, I. *et al.* **Specification, Implementation, and Deployment of COMPONENTS** – Communications of the ACM – October 2002/Vol. 45, No. 10.

Disponível em: <<http://www.idt.mdh.se/kurser/cd5490/2002/>>. Acesso em: 10 mar. 2005.

ÇURUKLU, B. **Towards a Visual Working Environment** - 2000

Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.

Acesso em: 09 fev. 2004.

DAINOW, B. **Intelligent Finance – XML Web Services In Action.**

Disponível em: <<http://www.xml.org/xml/intelligent-finance.pdf>>. Acesso em: 10 set. 2003.

DART, S. **Concepts in Configuration Management Systems**

Disponível em: <[ftp://ftp.sei.cmu.edu/pub/case-env/config\\_mgt/papers/cm\\_concepts.pdf](ftp://ftp.sei.cmu.edu/pub/case-env/config_mgt/papers/cm_concepts.pdf)>.

Acesso em: 24 fev. 2005.

DIAS, J. **Aplicando a Orientação a Objetos para o Desenvolvimento de Sistemas de Captação, Localização e Apoio à Qualificação de Produtos Seminovos**. 2002. Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, 2002.

DOUGHTY, F. **CORBA ORB Architecture** – 1996.

Disponível em: <<http://www.mtl.mit.edu/CAPAM/memos/96-7/node6.html>>. Acesso em: 14 out. 2003.

ELENKO, M.; REINERTSEN, M. **XML & CORBA** [s.d.]

Disponível em: <<http://www.omg.org/library/adt.htm>>. Acesso em: 10 mar. 2003.

FERRARI, G. **Java RMI (2)** [s.d.].

Disponível em: <<http://www.di.unipi.it/~giangi/CORSI/COOP/LECTURES/jrmi2.pdf>>.

Acesso em: 21 out. 2003.

FLORIJN, G. **The quality of style** – 2002.

Disponível em: <<http://www.cs.uu.nl/docs/vakken/swa/20012002/Slides/SA-7-StylesAndQualities.pdf>>. Acesso em: 04 nov. 2003.

FONG, J.; HUI, R.; **Application of middleware in the three tier client/server database design methodology** – Journal of the Brazilian Computer Society – vol.6 n.1 Campinas July 1999 - ISSN 0104-6500.

Disponível em: <<http://www.scielo.br>>. Acesso em: 06 ago. 2003.

GISOLFI, D. **Web services architect, Part3: Is Web services the reincarnation of CORBA?** – Solutions Architect, IBM Start Emerging Technologies – July 2001.

Disponível em: <<http://www-106.ibm.com/developerworks/library/ws-arc3/index.html>>.

Acesso em: 22 abr. 2004.

GRUNDGEIGER, D.; GOLDADE, A.; FUGMAN, V. **SQL and XML: to Invoke and Return Stored Procedures Over the Web** – MSDN Magazine, August 2001.



Disponível em: <<http://msdn.microsoft.com/msdnmag/issues/01/08/XMLSQL/default.aspx>>.  
Acesso em: 26 fev. 2003.

GURP, J. **Role Based Component Engineering** – University Karlskrona/Ronneby – 2000.  
Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.  
Acesso em: 09 fev. 2004.

HENRY, K. **Distributed Computation with Java Remote Method Invocation** – 21.01.2001.  
Disponível em: <<http://www.acm.org/crossroads/xrds6-5/ovp65.html>>. Acesso em: 22 out. 2003.

HO, S.; LAU, K.; LUDERS, F. **Specification of Components**  
Disponível em: <<http://www.idt.mdh.se/kurser/cd5490/2002/lectures/CBSE-Component Specification.pdf>>. Acesso em: 12 dez. 2003.

IBM. IBM – **Software – Tivoli – Desempenho e disponibilidade**  
Disponível em: <<http://www.ibm.com/br/products/software/tivoli/pa/>>. Acesso em: 17 ago. 2004.

INFOWORLD MEDIA GROUP **Application Server Techindex** – 2004  
Disponível em: <[http://www.infoworld.com/techindex/application\\_servers.html](http://www.infoworld.com/techindex/application_servers.html)>. Acesso em: 24 fev. 2005.

JAGANNATHAN, R.; SIVILOTTI, P. **Increasing Client-Side Confidence in Remote Component Implementations** – [s.d.] - The Ohio State University – Columbus, Ohio, USA.  
Disponível em: <<http://portal.acm.org/citation.cfm?id=503217>>. Acesso em: 24 fev. 2005.

INTERLEAF, A Sponsor Member of OÁSIS – **XML to Work: Advantages of Content Management** – [s.d.]  
Disponível em: <[http://www.xml.org/xml/putting\\_xml\\_to\\_work.shtml](http://www.xml.org/xml/putting_xml_to_work.shtml)>. Acesso em: 15 dez. 2003.

KIZILTAN, Z.; JONSSON T.; HNICH, B. **On the Definition of Concepts in Component Based Software Development** – 2000.  
Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.  
Acesso em: 09 fev. 2004.

LARSSON, M. **The Different Aspects of Component Based Systems** – 2000.  
Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.  
Acesso em: 09 fev. 2004.

LARSSON, M; CRNKOVIC, I. **Component Configuration Management**.  
Disponível em: <<http://www.mrtc.mdh.se/publications/0236.pdf>>. Acesso em: 19 set. 2003.

LARSSON, T.; SANDBERG, M. **Building Flexible Components Based on Design Patterns** – Malardalen University – 2000.  
Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.  
Acesso em: 09 fev. 2004.

LUDERS, F. **Architectural Styles in Component-Based Software Engineering** – 2000.

Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.

Acesso em: 09 fev. 2004.

MCCARTY, B; CASSIDY-DORION, L.; **JAVA Distributed Objects** – 1999 – ISBN 0-672-31537-8

MICROSOFT **DCOM Architecture White Paper** – 1998.

Disponível em: <<http://www.microsoft.com>>. Acesso em: 17 out. 2003.

OBJECT MANAGEMENT GROUP – OMG – **ORB Basics** – 1997-2004.

Disponível em: <[http://www.omg.org/gettingstarted/orb\\_basics.htm](http://www.omg.org/gettingstarted/orb_basics.htm)>. Acesso em: 17 jun. 2004.

ORFANI, R.; HARKEY, D. **Client/Server Programming with Java and CORBA**. Second Edition. 1998 – ISBN 0-471-24578-X

ORFANI, R.; HARKEY, D.; EDWARDS, J. **Client /Server Survival Guide** – Third Edition. 1999 – ISBN 0-471-31615-6

PEREIRA FILHO, S. F. **Avaliação de ambientes servidores para agentes móveis**. São Carlos – SP, 2001.

Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08022002-173805/publico/dissertacaostenio.pdf>>. Acesso em: 17 fev. 2003.

PLASIL, F.; STAL, M. **An Architectural View of Distributed Objects and Components in CORBA, Java RMI, and COM/DCOM** – Springer 1998.

Disponível em: <<http://nenya.ms.mff.cuni.cz/~plasil/publications.html>>. Acesso em: 19 fev. 2004.

PODMORE, R. et al. **Common Information Model – A Developer's Perspective** – Proceedings of the 32<sup>nd</sup> Hawaii International Conference on System Sciences -1999.

Disponível em:

<<http://csdl.computer.org/comp/proceedings/hicss/1999/0001/03/00013016.PDF>>. Acesso em: 27 fev. 2005.

PODMORE, R.; ROBINSON M. **Introduction to CIM** – November 5, 2000.

Disponível em: <[http://www.epri.com/attachments/292151\\_CimIntroduction.pdf](http://www.epri.com/attachments/292151_CimIntroduction.pdf)>. Acesso em: 07 jun. 2005.

Proceedings of ACM Multimedia 2001 Workshops – **International Workshop On Multimedia Middleware**. – October 5, 2001 – Ottawa, Ontário, Canadá.

Disponível em: <<http://www.ifi.uio.no/~m3w/>>. Acesso em: 25 fev. 2005.

RAJ, G. **A Detailed Comparison of CORBA, DCOM and Java/RMI** - 1998.

Disponível em: <<http://my.execpc.com/~gopalan/misc/compare.html>>. Acesso em: 05 nov. 2003.

RIGHETTI, M. **Mundo Java: SOA Arquitetura orientada a serviços** – número 6 – ano I (2004).

Disponível em:

<[http://www.linuxmall.com.br/shop\\_image/product/da9cba6b4a6fe8c14dc1f5ecc3675429.jpg](http://www.linuxmall.com.br/shop_image/product/da9cba6b4a6fe8c14dc1f5ecc3675429.jpg)>. Acesso em: 25 fev. 2005.

ROBILLARD, M. **Separation of Concerns and Software Components** – Department of Computer Science University of British Columbia – Canada – 2000.

Disponível em:

<[http://www.idt.mdh.se/kurser/phd/CBSE/reports/Final\\_reports/report\\_06.pdf](http://www.idt.mdh.se/kurser/phd/CBSE/reports/Final_reports/report_06.pdf)>. Acesso em: 25 fev. 2005.

SALEMI, J.; **PC Magazine Guide to Client/Server Databases** – 1993 – ISBN 1-56276-070-X

SCHUR, S. G.; **The Database Factory** – ISBN 0-471-55844-3

STEÉN, H. **Make smaller components work** – 2000.

Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.

Acesso em: 09 fev. 2004.

STONEBRAKER, M.; **Too Much Middleware** – EECS Department – M.I.T.

Disponível em: <<http://www.acm.org/sigmod/record/issues/0203/industry-ms.pdf>>. Acesso em: 19 set. 2003.

SUN DEVELOPER NETWORK - **JAVA REMOTE METHOD INVOCATION** – 2002.

Disponível em: <[http://java.sun.com/marketing/collateral/rmi\\_ds.html](http://java.sun.com/marketing/collateral/rmi_ds.html)>. Acesso em: 22 out. 2003.

SVAHNBERG, M.; HOGSTROM, M. **Designing Components for Variability** – University of Karlskrona/Ronneby – 2000.

Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.

Acesso em: 09 fev. 2004.

SZYPERSKI, C. **Component Software Beyond Object Oriented Programming**, ACM Press Book, 1997 - ISBN: 0201178885

THOMPSON, C. *et al.* **Intermediary Architecture: Interposing Middleware Object Services between Web Client and Server** – 1999 ACM 0360-0300/99/1200.

Disponível em: <<http://delivery.acm.org>>. Acesso em: 06 ago. 2003.

THOMPSON, J. **Overview of Middleware Technology** – Palestra da Gartner Group **Conferência sobre a Integração de Aplicativos e Serviços na Web** – Março de 2003.

Disponível em: <[http://www.gartner.com/2\\_events/conferences/2003/mex14l/mex14l\\_1.jsp](http://www.gartner.com/2_events/conferences/2003/mex14l/mex14l_1.jsp)>.

Acesso em: 09 set. 2005.

\_\_\_\_\_. **Selecting an Integration Broker: The Big vs. the Little** – Palestra da Gartner Group **Conferência sobre a Integração de Aplicativos e Serviços na Web** – Abril de 2004.

Disponível em: <[https://www.g2r.com/2\\_events/conferences/2004/mex16l/mex16l\\_1a.jsp](https://www.g2r.com/2_events/conferences/2004/mex16l/mex16l_1a.jsp)>.

Acesso em: 09 set. 2005.

Tradução livre do **PMBOK 2000, V 1.0**, disponibilizada através da Internet pelo PMI MG em janeiro de 2002.

Disponível em: <<http://www.pmimg.org.br/>>. Acesso em: 11 mar. 2005.

VAALUNDI, T.; ERIKSSON, N. **Applying CBSE theory on corporate resources** – 2000

Disponível em: <<http://www.idt.mdh.se/~icc/articles/2000/CBSE-State-of-the-art.pdf>>.

Acesso em: 09 fev. 2004.

VAN VLIET, P.J.A. **Introduction to Middleware – 2** – University of Nebraska at Omaha - [s.d.].

Disponível em: <<http://www.istis.unomaha.edu/isqa/vanvliet/arch/archpdf/wk10-middlewareintro2-act-4.PDF>>. Acesso em: 15 nov. 2003.

VAROTO, A. **Visões em arquitetura de Software**. IME-USP, São Paulo.

Disponível em: <<http://www.ime.usp.br/dcc/posgrad/teses/ane.pdf>>. Acesso em: 11 fev. 2003.

WIEDERHOLD, G. **Value-added Middleware: Mediators** – March 1998.

Disponível em: <<http://www-db.stanford.edu/pub/gio/1998/dbpd.html>>. Acesso em: 14 abr. 2003.

\_\_\_\_\_. **Mediation to Deal with Heterogeneous Data Sources** – Jan. 1999.

Disponível em: <<http://www-db.stanford.edu/pub/gio/1999/Interopdocfigs.html>>. Acesso em: 14 abr. 2003.

ZEID, A. **Key Components of Agent-based Development**. Computer Science Department, The American University in Cairo.

Disponível em: <<http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-59/13ZeidPosition.pdf>>. Acesso em: 19 dez. 2002.

ZHANG, C.; JACOBSEN, H. **Quantifying Aspects in Middleware Plata forms** – ACM 2003 1-58113-660-9/03/002.

Disponível em: <<http://portal.acm.org>>. Acesso em: 06 ago. 2003.

## Glossário

| Termo                                    | Descrição   |
|--|---|
| <i>Application Programming Interface</i> | É um conjunto de chamadas de procedimentos utilizados pelos programas para utilizarem o mediador.   |
| <i>Browser</i>                           | <i>Browser</i> ou navegador é um programa utilizado para visualizar páginas <i>Web</i> (exemplo: <i>Navigator</i> , <i>Internet Explorer</i> ).                             |
| Camada mediadora                         | Sinônimos: camada de lógica de aplicação ou camada de lógica de negócio.  |
| <i>Driver</i> do BD                      | É um <i>software</i> que atua como um mediador entre o aplicativo e o banco de dados. O <i>software</i> permite as conexões e manipulações sobre o banco de dados.          |
| GUI                                      | É um termo geral utilizado para definir uma classe de sistemas operacionais ou ambientes operacionais que promovem a interface do usuário sobre gráficos ao invés de texto. |
| Interface                                | A interface é alguma coisa necessária para os equipamentos ou programas criarem comunicação entre diferentes sistemas ou programas.   |
| Interface dos componentes                | Contrato ou acordo em relação aos métodos e os parâmetros passados por estes métodos.   |
| Java                                     | Linguagem de programação orientada a objetos criada pela empresa “ <i>Sun Microsystems</i> ”.   |
| <i>Skeleton</i>                          | Converte mensagens em chamadas de operações.  |
| <i>Statefull</i>                         | Mantém as informações de estado durante uma sessão.   |
| <i>Stateless</i>                         | Mantém as informações de estado durante uma transação.  |
| <i>Stored Procedure</i>                  | É um dos objetos que constitui um SGBDR. É um objeto que contém a lógica de programação e pode ser executado.   |
| <i>Stub</i>                              | Converte chamada de operações em mensagens.   |
| Sun Microsystems                         | Empresa fabricante de <i>software</i> e <i>hardware</i> tais como: o sistema operacional <i>Solaris</i> e linguagem de programação Java.                                    |
| XML                                      | Linguagem usada para a apresentação – descrição dos dados e está focada em informar em que consiste um dado?  |

## Apêndice A – Diagramas de Caso de Uso.

Com o intuito de organizar melhor o trabalho os Casos de Uso são apresentados a seguir em quatro itens de acordo com os requisitos funcionais apresentados no capítulo 3. Sendo assim, os itens a seguir constituem os seguintes grupos: Grupo de registro e manutenção (fusão do 3.2-a - Grupo de registro com o 3.2-b - Grupo de manutenção), Grupo de controle, Grupo de alertas e o Grupo das consultas ou relatórios.

Todos os casos de uso utilizam o mesmo modelo de apresentação, que segue com o título do caso de uso, os atores que participam do caso de uso, os requisitos funcionais viabilizados pelo caso de uso, um texto descritivo sobre o caso de uso, detalhes de como é iniciado o caso de uso, o fluxo de eventos contemplados pelo caso de uso e, finalmente, uma figura que sintetiza visualmente o caso de uso.

### 1) Grupo de registro ou Grupo de manutenção

- a) Título: Cadastramento e manutenção do acervo de serviços.  
Ator: administrador de serviços.

Viabiliza os requisitos funcionais: 3.2-a e 3.2-b.

Descrição: O administrador cadastra os serviços que são ofertados, privilegiando as informações que os identificam, tais como, as datas que registram o início e o término de sua “vida” e o seu período de utilização durante um dia. O administrador também pode alterar, consultar ou eliminar essas informações. As ações aplicadas no gerenciador de serviços se estendem ao servidor de aplicação. A frequência de utilização dessa atividade dependerá da necessidade de publicação de novos serviços, da manutenção dos já existentes, de mudanças de diretrizes ou da eliminação dos mesmos.

Início: Esse caso de uso começa quando o administrador recebe a incumbência de implantar ou re-implantar um serviço, ou quando é necessário alterar alguma configuração de um serviço ou do ambiente.

Fluxo de eventos:

- Sequência Padrão:

| # | Ator   | Sistema  |
|---|--|--|
| 1 | Selecionar no menu a opção cadastramento de serviços.  | A tela de serviços é apresentada ao administrador.                         |
| 2 | Digitar informações que identificam o serviço.   | Se o serviço já estiver cadastrado, apresentar as informações registradas. |
| 3 | Digitar informações do serviço como descrição, versão, datas, interface, componentes utilizados, entre outros. | Nenhuma.   |
| 4 | Selecionar o botão Salvar.   | Registrar informações indicadas.   |
| 5 | Fim.   |  |

• Sequência Alternativa 1:

| # | Ator  | Sistema   |
|---|---|---|
| 1 | Digitar informações que identificam o serviço.  | Se o serviço já estiver cadastrado, apresentar informações registradas. |
| 2 | Selecionar o botão Alterar.   | Os campos são liberados para a alteração.                               |
| 3 | Digitar alterações do serviço como descrição, versão, datas, interface, componentes utilizados, entre outros. | Nenhuma.  |
| 4 | Selecionar o botão Salvar.  | Registrar informações alteradas.  |
| 5 | Fim.  |   |

• Sequência Alternativa 2:

| # | Ator   | Sistema   |
|---|--|---|
| 1 | Digitar informações que identificam o serviço. | Se o serviço já estiver cadastrado, apresentar informações registradas.       |
| 2 | Selecionar o botão Excluir.                    | Informar para registro o motivo da exclusão.<br>Excluir o serviço cadastrado. |
| 3 | Fim.   |   |

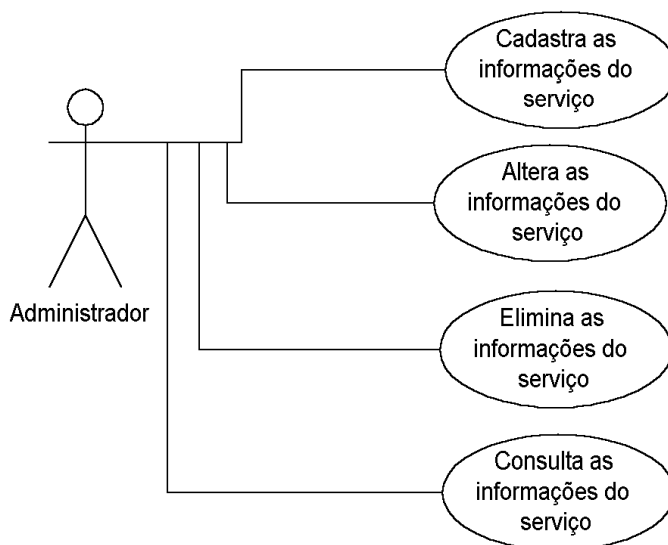


Figura A.1 – Cadastramento e manutenção do acervo de serviços.

b) Título: Cadastramento e administração dos serviços adquiridos de terceiros.

Ator: administrador de serviços.

Viabiliza os requisitos funcionais: 3.2-a e 3.2-b.

Descrição: O administrador também pode gerenciar os serviços adquiridos de terceiros (fornecedores externos à empresa) através do registro de informações sobre esses fornecedores e de datas que tratam do vencimento dos contratos de suporte com os mesmos.

Início: Esse caso de uso começa quando o administrador recebe a incumbência de registrar e controlar os serviços fornecidos por terceiros.

Fluxo de eventos:

• Sequência Padrão:

| # | Ator   | Sistema  |
|---|--|--|
| 1 | Selecionar no menu a opção cadastramento de serviços.  | A tela de serviços é apresentada ao administrador.                         |
| 2 | Digitar informações que identificam o serviço.   | Se o serviço já estiver cadastrado, apresentar as informações registradas. |
| 3 | Selecionar o botão Alterar.  | Os campos são liberados para a alteração.                                  |
| 4 | Digitar informações do serviço como datas, número de contrato e as informações dos fornecedores. | Nenhuma.   |
| 5 | Selecionar o botão Salvar.   | Registrar informações indicadas.   |
| 6 | Fim.   |  |

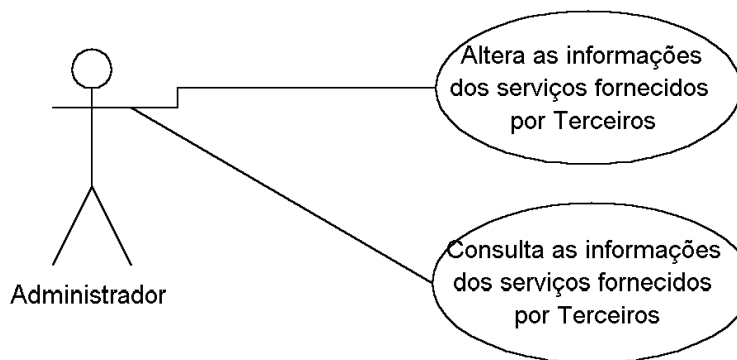


Figura A.2 – Cadastramento de serviços adquiridos de terceiros.

c) Título: Cadastramento e manutenção das informações sobre a interface dos serviços.

Ator: administrador de serviços.

Viabiliza os requisitos funcionais: 3.2-a e 3.2-b.

Descrição: O administrador cadastra as especificações da interface do serviço. Essa ação ocorre durante o cadastramento de um novo serviço ou quando houver uma modificação da interface do serviço em questão.

Início: Esse caso de uso começa quando o administrador recebe a incumbência de implantar ou re-implantar um serviço ou quando necessita alterar alguma configuração da interface de um serviço.

Fluxo de eventos:

• Sequência Padrão:

| # | Ator  | Sistema   |
|---|---|---|
| 1 | Selecionar no menu a opção cadastramento de interface do serviço. | A tela de interface do serviço é apresentada ao administrador.                          |
| 2 | Digitar informações que identificam o serviço.                    | Se o serviço já estiver cadastrado, apresentar as informações registradas da interface. |



Continuação da tabela: Seqüência Padrão.

|   |  |                                  |
|---|--|----------------------------------|
| 3 | Digitar informações da interface do serviço como argumento, tipo do dado, seqüência do <i>layout</i> da interface, entre outros. | Nenhuma.                         |
| 4 | Selecionar o botão Salvar.   | Registrar informações indicadas. |
| 5 | Fim.   |                                  |

• Seqüência Alternativa 1:

| # | Ator  | Sistema   |
|---|---|---|
| 1 | Digitar informações que identificam o serviço.  | Se o serviço já estiver cadastrado, apresentar as informações registradas da interface. |
| 2 | Selecionar o botão Alterar.   | Os campos são liberados para a alteração.   |
| 3 | Digitar alterações da interface do serviço como argumento, tipo do dado, seqüência do <i>layout</i> da interface, entre outros. | Nenhuma.  |
| 4 | Selecionar o botão Salvar.  | Registrar informações alteradas.  |
| 5 | Fim.  |   |

• Seqüência Alternativa 2:

| # | Ator   | Sistema   |
|---|--|---|
| 1 | Digitar informações que identificam o serviço. | Se o serviço já estiver cadastrado, apresentar as informações registradas da interface. |
| 2 | Selecionar o botão Excluir.                    | Informar para registro o motivo da exclusão.<br>Excluir a interface do serviço.         |
| 3 | Fim.   |   |

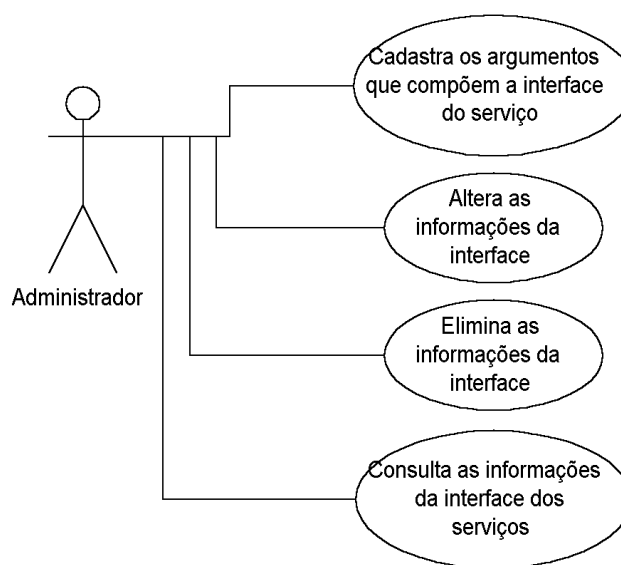


Figura A.3 – Cadastramento e manutenção da interface do serviço.

d) Título: Cadastramento e manutenção dos componentes utilizados pelos serviços.

Ator: administrador de serviços.

Viabiliza os requisitos funcionais: 3.2-a e 3.2-b.

Descrição: O administrador cadastra os elementos exigidos para o funcionamento do serviço, que podem ser outros serviços, componentes mais simples ou *drivers*.

Início: Essa ação ocorre durante o cadastramento de um novo serviço, quando houver uma modificação do serviço em questão ou quando houver alguma alteração na infra-estrutura que provê os componentes dependentes.

Fluxo de eventos:

- Sequência Padrão:

| # | Ator  | Sistema  |
|---|---|--|
| 1 | Selecionar no menu a opção cadastramento dos componentes utilizados pelo serviço.   | A tela dos componentes utilizados pelo serviço é apresentada ao administrador.             |
| 2 | Digitar informações que identificam o serviço.  | Se o serviço já estiver cadastrado, apresentar as informações registradas dos componentes. |
| 3 | Digitar informações dos componentes utilizados pelo serviço como nome do componente, versão, local de instalação, entre outros. | Nenhuma.   |
| 4 | Selecionar o botão Salvar.  | Registrar informações indicadas.   |
| 5 | Fim.  |  |

- Sequência Alternativa 1:

| # | Ator  | Sistema  |
|---|---|--|
| 1 | Digitar informações que identificam o serviço.  | Se o serviço já estiver cadastrado, apresentar as informações registradas dos componentes. |
| 2 | Selecionar o botão Alterar.   | Os campos são liberados para a alteração.  |
| 3 | Digitar informações dos componentes utilizados pelo serviço como nome do componente, versão, local de instalação, entre outros. | Nenhuma.   |
| 4 | Selecionar o botão Salvar.  | Registrar informações alteradas.   |
| 5 | Fim.  |  |

- Sequência Alternativa 2:

| # | Ator   | Sistema   |
|---|--|---|
| 1 | Digitar informações que identificam o serviço. | Se o serviço já estiver cadastrado, apresentar as informações registradas dos componentes.      |
| 2 | Selecionar o botão Excluir.                    | Informar para registro o motivo da exclusão.<br>Excluir os componentes utilizados pelo serviço. |
| 3 | Fim.   |   |

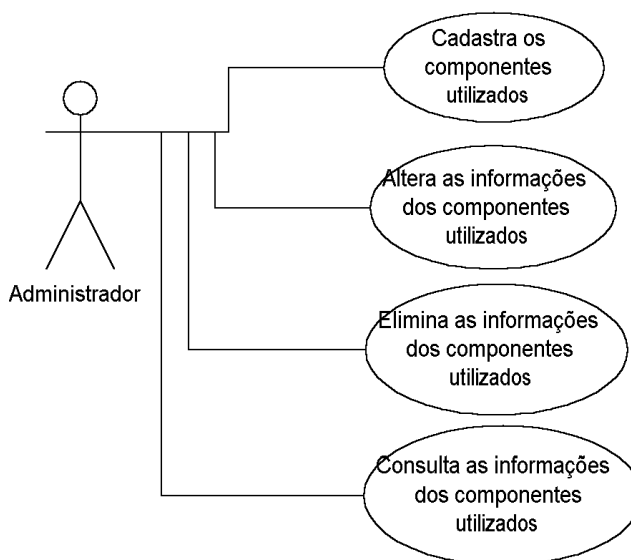


Figura A.4 – Cadastramento e manutenção dos componentes utilizados pelos serviços .

- e) Título: Cadastramento e manutenção dos problemas e soluções que ocorrem com os serviços.

Ator: administrador de serviços.

Viabiliza o requisito funcional: 3.2-b.

Descrição: O administrador registrará todos os problemas que ocorrem com o serviço durante o seu “tempo de vida” no ambiente de produção. O histórico do registro de problemas viabilizará uma base de informações para pesquisas do tipo histórica, de indisponibilidade do serviço prestado, de rastreamento e de anomalias. A expectativa de uso desta funcionalidade deveria ser baixa, mas é fundamental, pois todo *software*, como o próprio ambiente sistêmico, está sujeito a problemas.

Início: Esse caso de uso começa quando o administrador recebe como retorno dos usuários dos serviços, dos aplicativos ou do próprio sistema informes a respeito de erros.

Fluxo de eventos:

- Sequência Padrão:

| # | Ator  | Sistema  |
|---|---|--|
| 1 | Selecionar no menu a opção cadastramento de problemas com os serviços.  | A tela de registro de problemas é apresentada ao administrador.                                      |
| 2 | Digitar informações que identificam o serviço com o respectivo problema.  | Se o serviço com o respectivo problema já estiver cadastrado, apresentar as informações registradas. |
| 3 | Digitar informações do problema com o serviço como data, descrição, tipo, responsável pelo registro, criticidade, situação, entre outros. | Nenhuma.   |
| 4 | Selecionar o botão Salvar.  | Registrar informações indicadas.<br>Avisar o responsável do problema.                                |
| 5 | Fim.  |  |

## Sequência Alternativa 1:

| # | Ator  | Sistema  |
|---|---|--|
| 1 | Digitar informações que identificam o serviço com o respectivo problema.  | Se o serviço com o respectivo problema já estiver cadastrado, apresentar as informações registradas. |
| 2 | Selecionar o botão Alterar.   | Os campos são liberados para a alteração.  |
| 3 | Digitar informações do problema com o serviço como data, descrição, tipo, responsável pelo registro, criticidade, situação, entre outros. | Nenhuma.   |
| 4 | Selecionar o botão Salvar.  | Registrar informações alteradas.<br>Avisar o responsável do problema.                                |
| 5 | Fim.  |  |

## • Sequência Alternativa 2:

| # | Ator   | Sistema  |
|---|--|--|
| 1 | Digitar informações que identificam o serviço com o respectivo problema. | Se o serviço com o respectivo problema já estiver cadastrado, apresentar as informações registradas. |
| 2 | Selecionar o botão Excluir.  | Excluir o registro do problema do serviço.<br>Avisar o responsável do problema.                      |
| 3 | Fim.   |  |

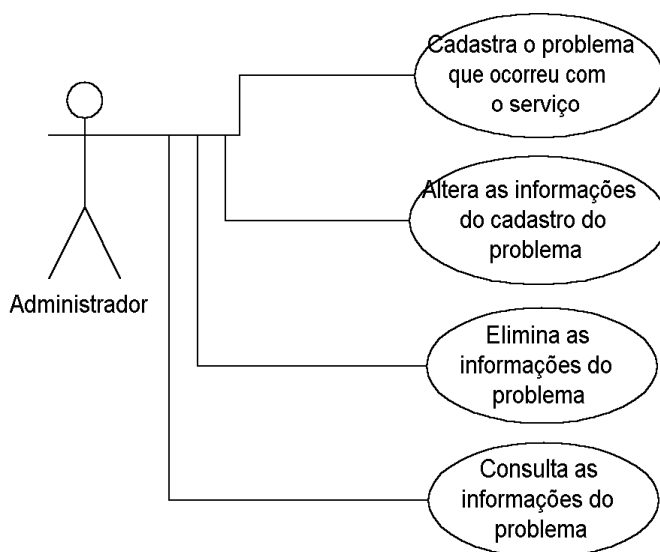


Figura A.5 – Cadastramento e manutenção dos problemas e soluções que ocorrem com os serviços.

- f) Título: Anulamento (ou Estorno) das últimas modificações (nos componentes utilizados, na interface e nas informações dos serviços) que ocorreram com os serviços.

Ator: administrador de serviços.

Viabiliza o requisito funcional: 3.2-b.

**Descrição:** O cadastramento das modificações (registro) é uma das funcionalidades mais importantes do projeto, portanto, para evitar a perda de confiabilidade da informação não há uma forma de editar uma modificação já gravada. Sendo assim, o administrador pode utilizar o registro de estorno para referenciar-se a alguma das últimas modificações já gravadas e, conseqüentemente, estorná-las (anulá-las). A atividade de estorno tem como exceções todas as transações que atuam com o servidor de aplicação, como, por exemplo, a alteração do estado do serviço. Além dessas, não é possível estornar uma modificação de estorno, ou seja, não é permitido executar um estorno sobre um outro estorno.

**Início:** O uso da funcionalidade do estorno geralmente ocorrerá para corrigir algum erro de digitação de algum lançamento de modificação ou em casos em que o lançamento em questão não mais procede.

Fluxo de eventos:

• Sequência Padrão:

| # | Ator   | Sistema   |
|---|--|---|
| 1 | Selecionar no menu a opção estorno de modificações.                              | A tela de estorno de modificações é apresentada ao administrador.                               |
| 2 | Digitar informações que identificam o serviço.                                   | Se o serviço já estiver cadastrado, apresentar a lista cronológica das modificações realizadas. |
| 3 | Identificar na lista cronológica de modificações o registro que sofrerá estorno. | A tela de estorno da referida modificação é apresentada ao administrador.                       |
| 4 | Digitar informações do estorno como data e descrição.                            | Nenhuma.  |
| 5 | Selecionar o botão Salvar.   | Registrar informações indicadas.  |
| 6 | Fim.   |   |

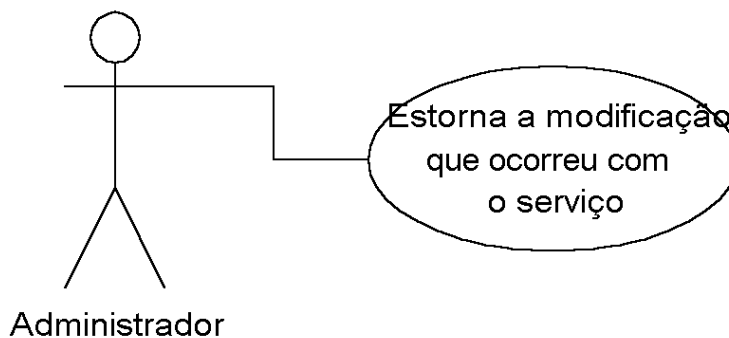


Figura A.6 – Estorno das modificações que ocorreram com os serviços.

g) Título: Alteração do estado do serviço.

Ator: administrador de serviços.

Viabiliza o requisito funcional: 3.2-b.

**Descrição:** A alteração do estado do serviço permite ativar, desativar e bloquear os serviços ofertados.

**Início:** Mediante alguma solicitação de mudança efetuada pelo responsável do serviço, o administrador pode, por exemplo, desativar, ativar ou bloquear um serviço e documentar as

informações pertinentes a essa ação, tais como, o nome do serviço que sofreu a ação, o motivo, a data, o horário, o solicitante e o executante.

Fluxo de eventos:

• Sequência Padrão:

| # | Ator   | Sistema   |
|---|--|---|
| 1 | Selecionar no menu a opção alteração do estado do serviço.   | A tela da alteração do estado do serviço é apresentada ao administrador.                    |
| 2 | Digitar informações que identificam o serviço.   | Se o serviço já estiver cadastrado, apresentar as informações da situação atual do serviço. |
| 3 | Digitar informações que registrem a ação de mudança de situação como a ação, o motivo, a data, o solicitante e o executante. | Nenhuma.  |
| 4 | Selecionar o botão Salvar.   | Registrar informações indicadas.  |
| 5 | Fim.   |   |

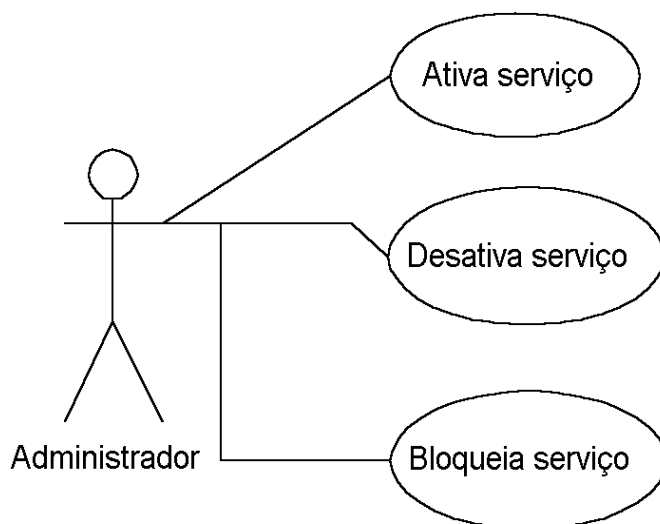


Figura A.7 – Alteração do Estado do serviço.

- h) Título: Cadastramento dos usuários responsáveis pelos serviços, pela homologação, pelos registros de modificações, pelos registros de problemas e soluções, pelos registros dos componentes utilizados pelos serviços e pesquisas no acervo.

Ator: administrador de serviços.

Viabiliza o requisito funcional: 3.2-a.

Descrição: O administrador cadastra as pessoas responsáveis pelas diversas manipulações que ocorrem com os serviços. O administrador também pode alterar, consultar e eliminar essas informações. A frequência de utilização dessa atividade depende da falta de um responsável para as necessidades de publicação de novos serviços, manutenção e gestão dos existentes e pesquisas no acervo.

Fluxo de eventos:

- Sequência Padrão:

| # | Ator  | Sistema  |
|---|---|--|
| 1 | Selecionar no menu a opção cadastramento de responsáveis.   | A tela de cadastramento de responsáveis é apresentada ao administrador.        |
| 2 | Digitar informações que identificam o responsável.  | Se o responsável já estiver cadastrado, apresentar as informações registradas. |
| 3 | Digitar informações do responsável como nome, telefone, <i>e-mail</i> e tipo de responsabilidade. | Nenhuma.   |
| 4 | Selecionar o botão Salvar.  | Registrar informações indicadas.   |
| 5 | Fim.  |  |

- Sequência Alternativa 1:

| # | Ator  | Sistema   |
|---|---|---|
| 1 | Digitar informações que identificam o responsável.  | Se o responsável já estiver cadastrado, apresentar informações registradas. |
| 2 | Selecionar o botão Alterar.   | Os campos são liberados para a alteração.                                   |
| 3 | Digitar informações do responsável como nome, telefone, <i>e-mail</i> e tipo de responsabilidade. | Nenhuma.  |
| 4 | Selecionar o botão Salvar.  | Registrar informações alteradas.  |
| 5 | Fim.  |   |

- Sequência Alternativa 2:

| # | Ator   | Sistema   |
|---|--|---|
| 1 | Digitar informações que identificam o responsável. | Se o responsável já estiver cadastrado, apresentar informações registradas.       |
| 2 | Selecionar o botão Excluir.                        | Informar para registro o motivo da exclusão.<br>Excluir o responsável cadastrado. |
| 3 | Fim.   |   |

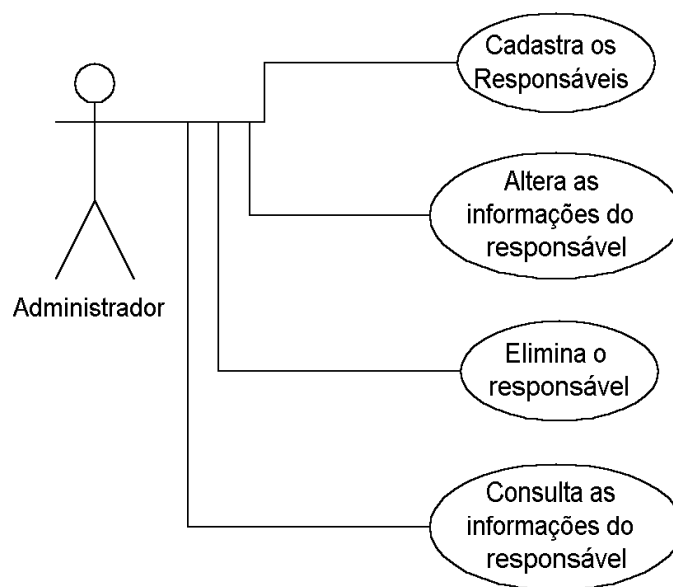


Figura A.8 – Cadastramento dos usuários responsáveis.

## 2) Grupo de Controle

- i) Título: Administração da data de expiração dos serviços.  
Atores: sistema e o administrador de serviços.

Viabiliza o requisito funcional: 3.2-c.

Descrição: O sistema auxilia na administração da data de expiração dos serviços adquiridos de terceiros, uma vez que esses, podem estar associados a uma data de expiração, por exemplo, devido a cláusulas do contrato de aquisição.

Início: O administrador ao ativar o sistema de gerenciamento de serviços é avisado, pelo próprio sistema, das datas expiradas e que estão a expirar.

Fluxo de eventos:

- Sequência Padrão:

| # | Ator   | Sistema  |
|---|--|--|
| 1 | Ativar o sistema de gerenciamento de serviços. | Em um dos setores da tela principal do sistema é apresentado uma sub tela com os contratos dos serviços que estão vencidos e a vencer. |
| 2 | Fim.   |  |



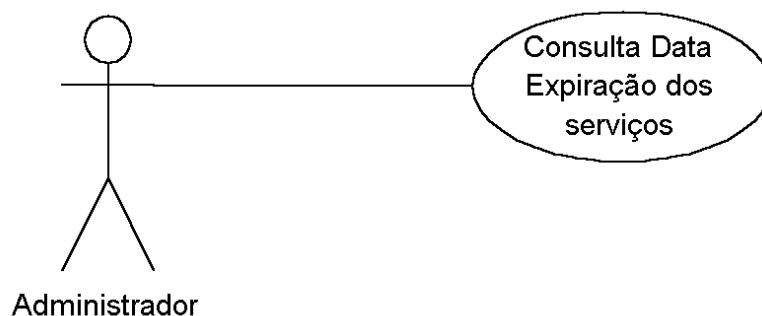


Figura A.9 – Administração da data de expiração dos serviços.

j) Título: Administração dos problemas não solucionados relativos aos serviços.

Atores: sistema e o administrador de serviços.

Viabiliza o requisito funcional: 3.2-c.

Descrição: Como podem ocorrer diversos problemas com os serviços e a solução dos mesmos precisa ser analisada, desenvolvida, testada e implantada, é importantíssimo ter uma forma de verificar quais problemas ainda precisam ser solucionados.

Início: O administrador ao ativar o sistema de gerenciamento de serviços é avisado, pelo próprio sistema, dos problemas não solucionados relativos aos serviços.

Fluxo de eventos:

• Sequência Padrão:

| # | Ator   | Sistema  |
|---|--|--|
| 1 | Ativar o sistema de gerenciamento de serviços. | Em um dos setores da tela principal do sistema é apresentado uma sub tela com os problemas não concluídos. |
| 2 | Fim.   |  |

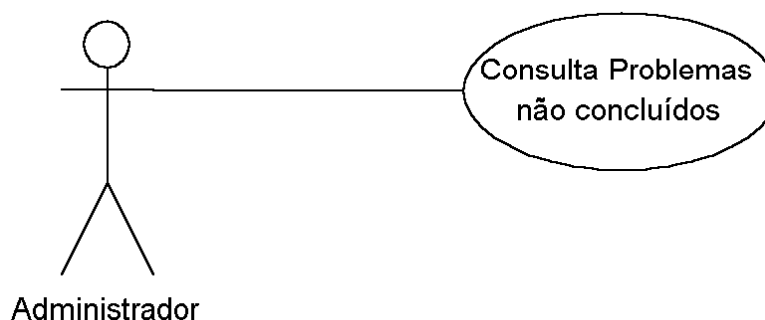


Figura A.10 – Administração dos problemas não solucionados relativos aos serviços.

k) Título: Administração do estado dos serviços.

Atores: sistema e o administrador de serviços.

Viabiliza o requisito funcional: 3.2-c.

Descrição: Como podem ocorrer diversos problemas com os serviços durante o uso é necessário consultar periodicamente o estado dos mesmos.

Início: O administrador ao ativar o sistema de gerenciamento de serviços é avisado, pelo próprio sistema, dos serviços que não estão funcionando.

Fluxo de eventos:

- Sequência Padrão:

| # | Ator   | Sistema   |
|---|--|---|
| 1 | Ativar o sistema de gerenciamento de serviços. | Em um dos setores da tela principal do sistema é apresentado uma sub tela com a lista dos serviços que não estão funcionando. |
| 2 | Fim.   |   |

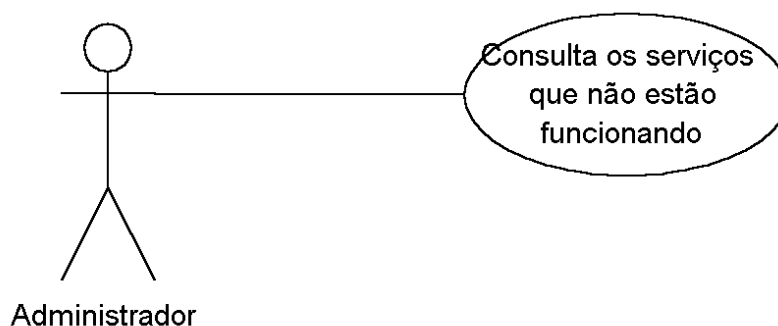


Figura A.11 – Administração do estado dos serviços.

### 3) Grupo de alertas (geração de avisos)

- 1) Título: Divulgação automática de informações sobre os serviços.

Atores: o sistema e os desenvolvedores de aplicativos ou serviços.

Viabiliza o requisito funcional: 3.2-d.

Descrição: O sistema divulgará automaticamente por intermédio de e-mail, a sinopse referente ao registro de novos serviços e a modificações de aspectos, de interface e dos componentes utilizados pelo serviço. O intuito dessa funcionalidade é informar cada vez mais os desenvolvedores sobre os serviços, sincronizando as informações, evitando assim erros e estimulando a reutilização.

Início: A divulgação tem início quando o administrador efetua uma gravação de aspectos do serviço, de informações sobre a interface e de informações sobre os componentes utilizados.

Fluxo de eventos:

- Sequência Padrão:

| # | Ator  | Sistema  |
|---|---|--|
| 1 | O administrador após inserir um novo registro ou alterar um registro existente clica no botão salvar. | O sistema após receber a instrução de inserir um novo registro ou alterar um registro existente aciona a rotina de divulgação.<br>A rotina de divulgação cria uma lista dos responsáveis que desejam ser informados e, conseqüentemente, envia os e-mails. |
| 2 | Fim.  |  |

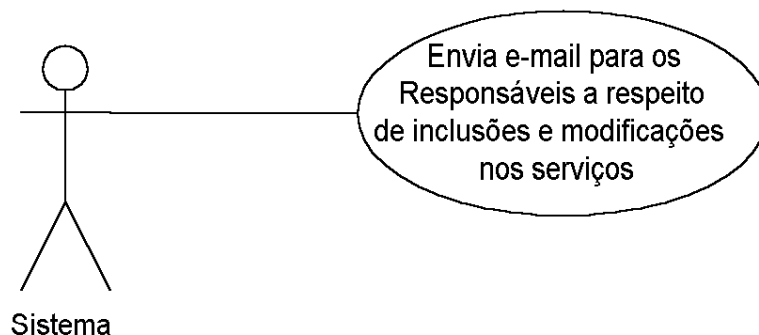


Figura A.12 – Divulgação automática de informações sobre os serviços.

#### 4) Grupo das consultas ou relatórios

- m) Título: Disponibilização da base de informações do gerenciador de serviços para pesquisas (do acervo, de configuração, de problemas, de soluções e de modificações) com finalidades operacionais, gerenciais, de rastreabilidade e reuso.

Atores: administrador de serviços e os desenvolvedores de aplicativos ou serviços.

Viabiliza o requisito funcional: 3.2-e.

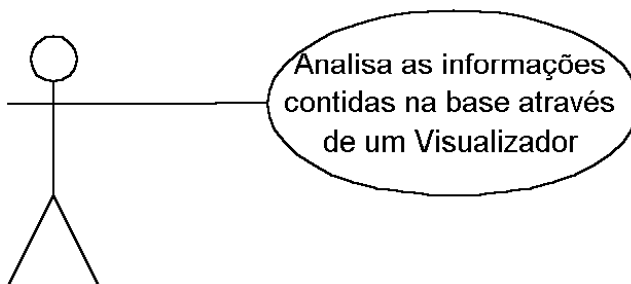
Descrição: O administrador precisa compor uma diversidade de consultas operacionais e gerenciais que precisam ser flexíveis, parametrizadas, re-classificadas de diversas maneiras e sumarizadas, ou seja, uma grande quantidade de análises que podem ser disponibilizadas para consumo através de consultas no vídeo ou relatórios impressos. A solução para atender a essa diversidade de consultas é disponibilizar o acesso de leitura à base de informações para o uso de ferramentas visualizadoras. Desse modo, em princípio, é possível obter do sistema, o seguinte conjunto de consultas:

- Estado dos serviços;
- Serviços por projeto;
- Serviços por versionamento;
- Os últimos 50 serviços instalados que estão em atividade.
- Responsáveis pelos serviços;
- Responsáveis pela homologação dos serviços;
- Horário viável para planejar a execução de manutenção do serviço, de um grupo de serviços ou da máquina;
- Problemas e modificações que ocorreram com os serviços;
- Problemas que estão pendentes;
- Problemas com as respectivas soluções;
- Re-implantações dos serviços;
- Percentual de re-implantações dos serviços por semana;
- Componentes utilizados pelos serviços.
- Estrutura da interface dos serviços.

Fluxo de eventos:

• Sequência Padrão:

| # | Ator  | Sistema  |
|---|---|--|
| 1 | Ativar alguma ferramenta visualizadora conectando-a ao banco de dados do sistema gerenciador de serviços. | A tela de elaboração de consultas ou relatórios é apresentada ao administrador ou desenvolvedor. |
| 2 | Digitar informações que identificam a consulta ou o relatório.  | Se a consulta ou o relatório já estiver cadastrado, apresenta as informações registradas.        |
| 3 | Digitar o <i>layout</i> da consulta ou do relatório.  | Nenhuma.   |
| 4 | Selecionar o botão executar.  | Executar a solicitação da consulta ou do relatório.  |
| 5 | Selecionar o botão Salvar.  | Registrar informações indicadas.   |
| 6 | Fim.  |  |



Administrador / Desenvolvedor

Figura A.13 – Análise das informações através de ferramentas externas.

## Apêndice B – Descrição dos atributos das Classes.

As tabelas abaixo apresentam as classes que compõem o projeto, seus atributos e a descrição do conteúdo da informação.

| Classe: SERVIÇO  |                                     |  |
|--|-------------------------------------|--|
| Identificador único do serviço                             | identificadorServiço                | Código interno que define uma identificação única para o serviço.  |
| Número da versão do serviço                                | numeroVersaoServiço                 | Número da versão do serviço.   |
| Nome do serviço  | nmServiço                           | Nome do serviço.   |
| Descrição do serviço                                       | descricaoServiço                    | Descrição detalhada do serviço.  |
| Identificador do responsável                               | identificadorResponsável            | Código interno que define uma identificação única para o responsável pelo serviço.   |
| Nome do Projeto que aloca o serviço                        | nmProjeto                           | Esse campo permite associar o serviço a um projeto. Exemplo, projeto datas, contém serviços de manipulação de datas, tais como, número de dias corridos entre datas, número de dias úteis entre datas, entre outros.   |
| Descrição da Plataforma operacional de execução do serviço | descricaoPlataformaOperacional      | Descrição da plataforma operacional que suportará o serviço.   |
| Nome do tipo de serviço                                    | nmTipoServiço                       | Alguns dos possíveis tipos de serviço: SP-Serviços gerais ligados aos negócios (cálculos de datas, verificação DAC, entre outros.); SN-Serviços específicos dos negócios (crédito de CC, débito de CC, entre outros.); SU-Serviços de suporte; SB-Serviços de acesso ao banco de dados; SC-Serviços de comunicação / transferência; SS-Serviços de segurança; SG-Serviços de gerência de serviços; SO-Serviços de gerenciamento do sistema operacional; SM-Serviços de monitoramento; SK-Serviços de armazenamento e recuperação de informações; SD-Serviços de depuração. |
| Data da inclusão   | DtInclusão                          | Data e hora de inclusão no registro do serviço na respectiva versão.   |
| Data da última modificação                                 | dtUltimaModificação                 | Data e hora da última modificação que ocorreu com o serviço na respectiva versão.  |
| Data de desativação  | dtDesativação                       | Data e hora de desativação do serviço – versão.  |
| Data de expiração do serviço                               | dtExpiração                         | Através dessa data é possível expirar a oferta do serviço sem inativar o mesmo.  |
| Identificador do responsável pela homologação              | identificadorResponsávelHomologação | Esse campo persiste a identificação do indivíduo que homologou o serviço.  |

Continuação da descrição da classe Serviço.

|  |                                 |   |
|--|---------------------------------|---|
| Data de homologação                            | dtHomologação                   | Data e hora em que ocorreu a homologação do serviço.  |
| Nome da razão social do Fornecedor             | nmRazãoSocialFornecedor         | Informações sobre o fornecedor do serviço.  |
| Número do contrato                             | numeroContrato                  | Número do contrato de suporte com o fornecedor.   |
| Data vencimento contrato de suporte            | dtVencimentoContrato            | Data de vencimento do contrato de suporte com o fornecedor. Se a informação ficar em branco não tem contrato.   |
| Nome da Linguagem de programação do serviço    | nmLinguagemProgramação          | Nome e versão da linguagem de programação do serviço (Java, Visual Basic, entre outros).  |
| Texto configuração                             | descriçãoConfiguração           | Campo texto que permite descrever a configuração do serviço. Esse campo também contém as configurações do ambiente que viabiliza a execução do serviço. |
| Nível de Criticidade                           | nívelCriticidade                | Nível de criticidade do serviço (Alta, Média, Baixa).   |
| Local da instalação do serviço                 | localInstalação                 | Nome do caminho físico da instalação do serviço.  |
| Data da instalação                             | dtInstalação                    | Data e hora de instalação do serviço no referido local.   |
| Período de atividade do serviço                | períodoAtividadeServiço         | Período do dia em que o serviço deve estar funcional.   |
| Quantidade em minutos do serviço fora ar       | qtdMinutosForaDoAr              | Valor em minutos que o serviço pode ficar fora do ar durante o horário de maior atividade, mas referente ao período de um dia.                          |
| Identificador único do serviço que o substitui | identificadorServiçoSubstituído | Identificador único do serviço. Quando um serviço é desativado pode ser substituído por outro serviço.  |
| Número da versão do serviço substituído        | numeroVersaoServiçoSubstituído  | Número da versão do serviço. Quando um serviço é desativado pode ser substituído por outro serviço.   |
| Status   | status                          | Possíveis situações das instâncias de dados (Ativo, Inativo, Não liberado para publicação, Em instalação, Em configuração, Em reconfiguração).          |

| Classe: INTERFACE DO SERVIÇO   |                      |   |
|--------------------------------|----------------------|---|
| Identificador único do serviço | identificadorServiço | Código interno que define uma identificação única para o serviço. |
| Número da versão serviço       | numeroVersaoServiço  | Número da versão do serviço.                                      |
| Nome do argumento              | nmArgumento          | Nome do argumento que compõe a interface de chamada do serviço.   |

Continuação da descrição da classe Interface.

|                                   |                            |  |
|-----------------------------------|----------------------------|--|
| Número de sequência do argumento  | numeroDaSequênciaArgumento | Número crescente que indica a posição do argumento.                                      |
| Nome do tipo de dado do argumento | nmTipoDeDadoArgumento      | O tipo de dado do argumento pode ser caractere, numérico, data, entre outros.            |
| Tamanho do argumento              | numeroTamanhoArgumento     | Tamanho em <i>bytes</i> do argumento.  |
| Descrição do argumento            | descriçãoArgumento         | Descrição sucinta do argumento.  |
| Direção                           | descriçãoDireção           | Os possíveis sentidos da direção do argumento ( <i>in</i> , <i>out</i> , <i>inout</i> ). |
| Observação sobre o argumento      | txtObservaçãoArgumento     | Texto livre que permite documentar qualquer observação a respeito do argumento.          |

| Classe: MODIFICAÇÕES NOS SERVIÇOS                        |                                    |   |
|--|------------------------------------|---|
| Identificador único do serviço                           | identificadorServiço               | Código interno que define uma identificação única para o serviço.   |
| Número da versão do serviço                              | numeroVersaoServiço                | Número da versão do serviço.  |
| Data da Modificação                                      | dtModificação                      | Data e hora de cadastramento da modificação no sistema.   |
| Identificador do responsável pela modificação            | identificadorResponsável           | Código interno que define uma identificação única para o responsável pela modificação. O objetivo desse atributo é registrar o responsável solicitante.   |
| Identificador do responsável que executará a modificação | identificadorResponsávelExecutante | Código interno que define uma identificação única para o responsável que executará a modificação solicitada.  |
| Nome do tipo de modificação                              | nmTipoModificação                  | Possíveis situações de modificação: Inclusão ou criação do serviço, Ativação serviço, Desativação serviço, Não publicar, Re-implantação, Modificação na especificação do serviço, Modificação dos componentes utilizados, Estorno de uma modificação, entre outros. |
| Descrição da modificação                                 | DescriçãoModificação               | Texto que permite descrever o assunto que envolve a necessidade da modificação.   |
| Data do estorno da modificação                           | dtEstorno                          | Data e hora em que ocorreu o registro de estorno de uma modificação. Campo obrigatório no caso de estorno.  |
| Quantidade minutos do serviço fora do ar                 | QtdMinutosForaDoAr                 | Valor em minutos que o serviço ficou fora do ar durante uma modificação no serviço.   |

| Classe: PROBLEMAS COM OS SERVIÇOS          |                                  |  |
|--|----------------------------------|--|
| Identificador único do serviço             | IdentificadorServiço             | Código interno que define uma identificação única para o serviço.  |
| Número da versão serviço                   | NumeroVersaoServiço              | Número da versão do serviço.   |
| Data de registro do problema               | DtRegistroProblema               | Data e hora de registro do problema no sistema.  |
| Nome do tipo do problema                   | nNmTipoProblema                  | O tipo do problema permite classificar o problema, por exemplo, conflito com os componentes utilizados, travamento, <i>login</i> de serviço, entre outros. |
| Data da constatação do problema            | dtConstataçãoProblema            | Data e hora provável da constatação do problema.   |
| Descrição do problema                      | descricaoProblema                | Descrição do problema.   |
| Identificador do responsável pelo problema | identificadorResponsavelProblema | Código interno que define uma identificação única para o responsável pelo registro do problema.  |
| Data de encerramento do problema           | dtEncerramentoProblema           | Data e hora do encerramento do problema.   |
| Descrição da solução                       | descricaoSolução                 | Descrição da solução do problema.  |
| Identificador do responsável pela solução  | identificadorResponsavelSolução  | Código interno que define uma identificação única para o responsável pelo registro da solução.   |
| Quantidade de minutos de indisponibilidade | qtdMinutosIndisponibilidade      | Valor em minutos que o serviço ficou fora do ar durante um problema que o impediu de agir.   |
| Nível de Criticidade                       | nívelCriticidade                 | Nível de criticidade do problema (Alta, Média, Baixa).   |
| Status                                     | status                           | Situação do problema (Em aberto, Em andamento, Concluído).   |

| Classe: RESPONSÁVEL PELAS AÇÕES NO SISTEMA |                          |   |
|--|--------------------------|---|
| Identificador do responsável pelo serviço  | identificadorResponsavel | Código interno que define uma identificação única para o responsável. |
| Data de registro                           | dtRegistro               | Data e hora de registro do responsável no sistema.                    |
| Nome do responsável                        | nmResponsavel            | Nome do responsável.  |
| E-mail do responsável                      | nmEmailResponsavel       | Endereço eletrônico do responsável.                                   |
| Telefone                                   | numTelefone              | Número do telefone do responsável.                                    |



Continuação da classe Responsável.

| Tipo de responsabilidade  | nmTipoResponsabilidade | Possíveis tipos de responsabilidade (Administrador, Desenvolvedor, Homologador).  |
|---|------------------------|---|
| Enviar e-mail informando sobre as modificações e inclusões dos serviços | idEnviarEmail          | Habilita o envio de e-mail para o responsável quando ocorrer alguma inclusão ou modificação nos aspectos, na interface e nos componentes utilizados pelo serviço. |

## Apêndice C – Descrição dos métodos das classes.

As tabelas a seguir apresentam em detalhes as funções (métodos) que participam dos diagramas de classes da figura 3.4 e 3.5.

| Classe: Serviço                                 |  |
|---|--|
| registrar dados do serviço()                    | Inclui ou altera os dados do serviço.                              |
| registrar dados dos componentes utilizados()    | Inclui ou altera os dados dos componentes utilizados pelo serviço. |
| registrar exclusão do serviço()                 | Efetua a exclusão do serviço.                                      |
| registrar exclusão dos componentes utilizados() | Efetua a exclusão dos componentes utilizados.                      |
| verificar serviço cadastrado()                  | Consulta o serviço para verificar se está cadastrado.              |
| verificar componentes utilizados()              | Consulta os componentes utilizados pelo serviço.                   |
| verificar estado serviço()                      | Consulta o estado do serviço.                                      |
| verificar serviços vencidos()                   | Consulta os serviços que possuem a data de expiração vencida.      |
| verificar serviços a vencer()                   | Consulta os serviços que possuem a data de expiração a vencer.     |
| listar serviços()                               | Lista os serviços cadastrados.                                     |

| Classe: Interface do serviço      |  |
|-----------------------------------|--|
| registrar dados da interface()    | Inclui ou altera a especificação do serviço.                           |
| registrar exclusão da interface() | Exclui a especificação do serviço.                                     |
| verificar interface cadastrada()  | Consulta a especificação do serviço para verificar se está cadastrada. |

| Classe: Modificações nos serviços |   |
|-----------------------------------|---|
| registrar modificação()           | Inclui uma nova modificação decorrente de alguma inclusão, alteração ou exclusão do serviço, da interface, dos componentes utilizados ou da alteração do estado do serviço. |
| registrar dados do estorno()      | Estorna uma modificação que já ocorreu.   |
| verificar modificações()          | Consulta as modificações de um serviço.   |
| verificar últimas modificações()  | Consulta as últimas modificações de um serviço.   |

| Classe: Problemas com os serviços    |   |
|--------------------------------------|---|
| registrar dados do problema()        | Inclui um novo problema ou altera as informações registradas de algum problema que ocorreu com um serviço. Entre os diversos tipos de alteração do registro de problema a principal é a solução do mesmo. |
| registrar exclusão do problema()     | Exclui o registro de problema de um serviço.  |
| verificar problema cadastrado()      | Consulta o problema de um serviço.  |
| verificar problemas não concluídos() | Consulta os problemas que não estão concluídos.   |

| Classe: Responsável pelas ações no sistema |  |
|--|--|
| registrar dados do responsável()           | Inclui ou altera as informações dos usuários responsáveis por administrar, por elaborar e por alterar os serviços. |
| registrar exclusão do responsável()        | Exclui um determinado responsável.   |
| verificar responsável cadastrado()         | Consulta as informações de um determinado responsável.   |
| listar responsáveis()                      | Lista os responsáveis que foram cadastrados.   |
| avisar responsável()                       | Envia uma mensagem para o endereço eletrônico cadastrado.  |

| Classe: Servidor de aplicação           |  |
|---|--|
| atualizar dados no servidor aplicação() | Através de uma API o sistema gerenciador de serviços pode atualizar o dados no ambiente do servidor de aplicação.                      |
| Verificar estado serviço()              | Através de uma API o sistema gerenciador de serviços pode verificar o estado dos serviços dentro do ambiente do servidor de aplicação. |

## Apêndice D – Diagramas de Seqüência.

Com o intuito de organizar melhor a elaboração do trabalho, os diagramas de seqüência são apresentados a seguir em três grupos de acordo com os requisitos funcionais apresentados no capítulo 3. Os grupos contemplados são: Grupo de registro e manutenção (fusão do 3.2-a - Grupo de registro com o 3.2-b - Grupo de manutenção), Grupo de controle e o Grupo das consultas ou relatórios.

Os itens que compõem os grupos seguem a mesma seqüência dos Casos de Uso do apêndice A, entretanto são apresentados dois diagramas para os casos de uso que tratam do tema cadastramento para destacar, além das operações de inclusão e alteração, a operação de exclusão.

Todos os diagramas de seqüência utilizam o mesmo modelo de apresentação, que segue com o título do caso de uso e a figura que sintetiza visualmente as instâncias das classes sob um determinado fluxo, ou seja, o diagrama de seqüência.

### 1) Grupo de registro ou Grupo de manutenção

#### a) Caso de Uso: Cadastramento de Serviços.

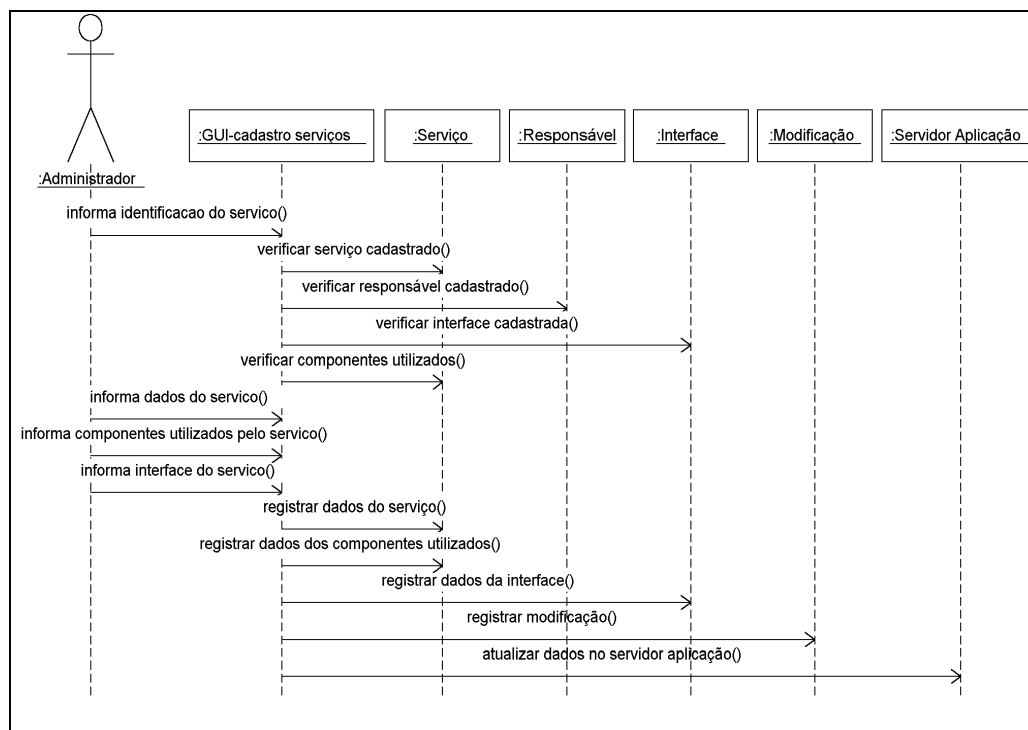


Figura D.1 – Diagrama de Seqüência - Cadastramento de Serviços.

### Caso de Uso: Cadastramento de Serviços – Exclusão .

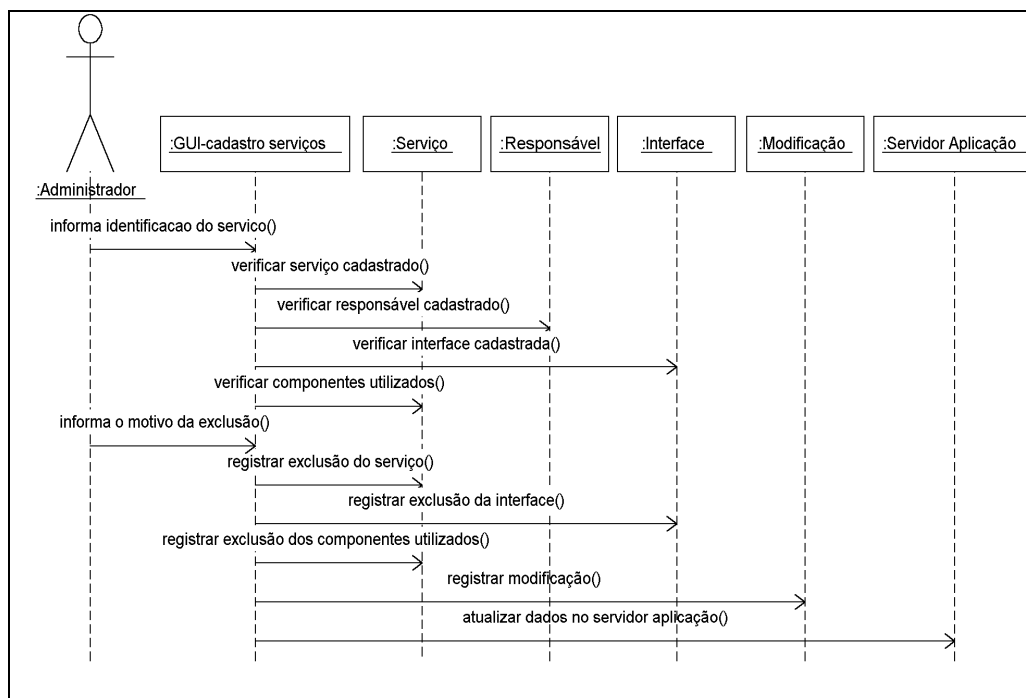


Figura D.2 – Diagrama de Seqüência – Cadastramento de Serviços – Exclusão.

### b) Caso de Uso: Cadastramento de Serviços de terceiros.

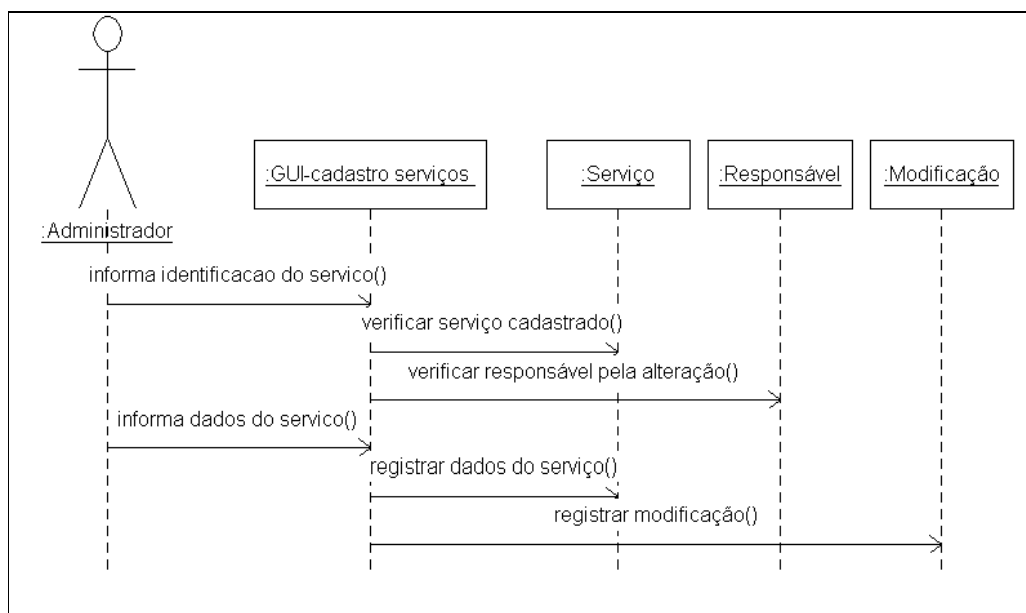


Figura D.3 – Diagrama de Seqüência – Cadastramento de Serviços de terceiros.

## c) Caso de Uso: Cadastramento da interface do serviço.

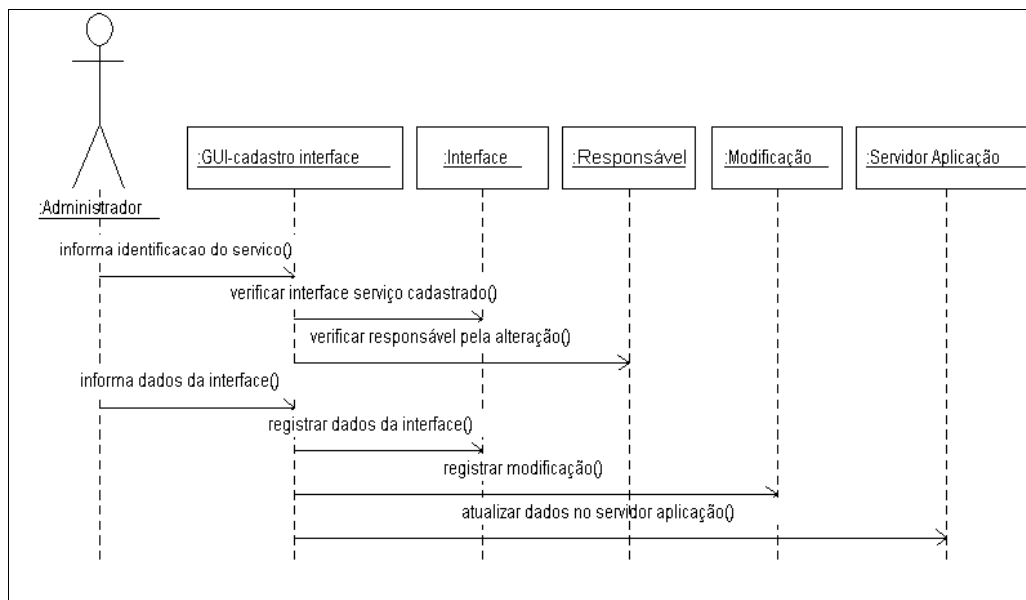


Figura D.4 – Diagrama de Seqüência – Cadastramento da interface do serviço.

## Caso de Uso: Cadastramento da interface do serviço - Exclusão.

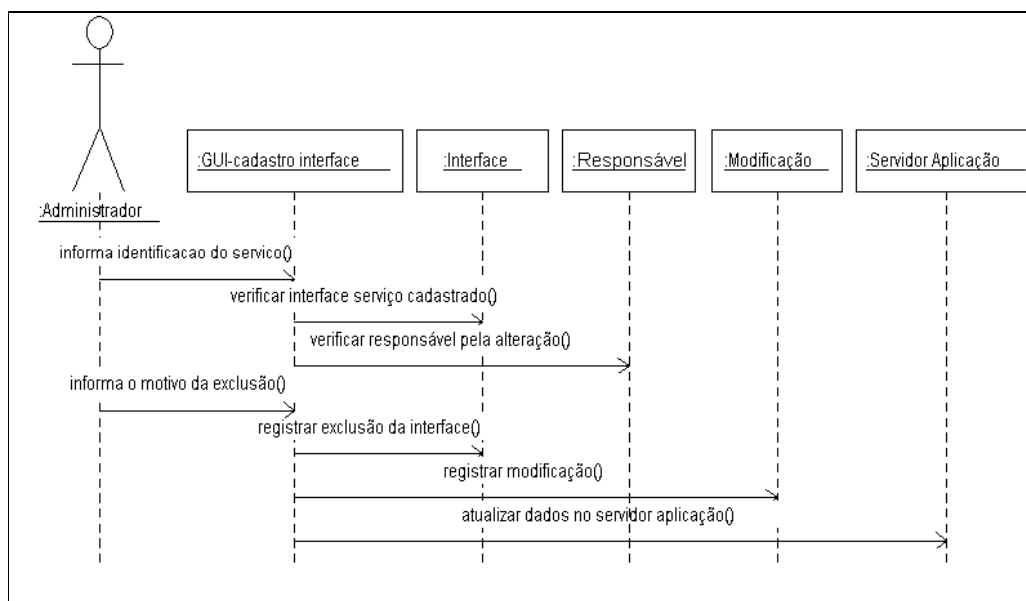


Figura D.5 – Diagrama de Seqüência – Cadastramento da interface do serviço – Exclusão.

d) Caso de Uso: Cadastramento dos componentes utilizados pelo serviço.

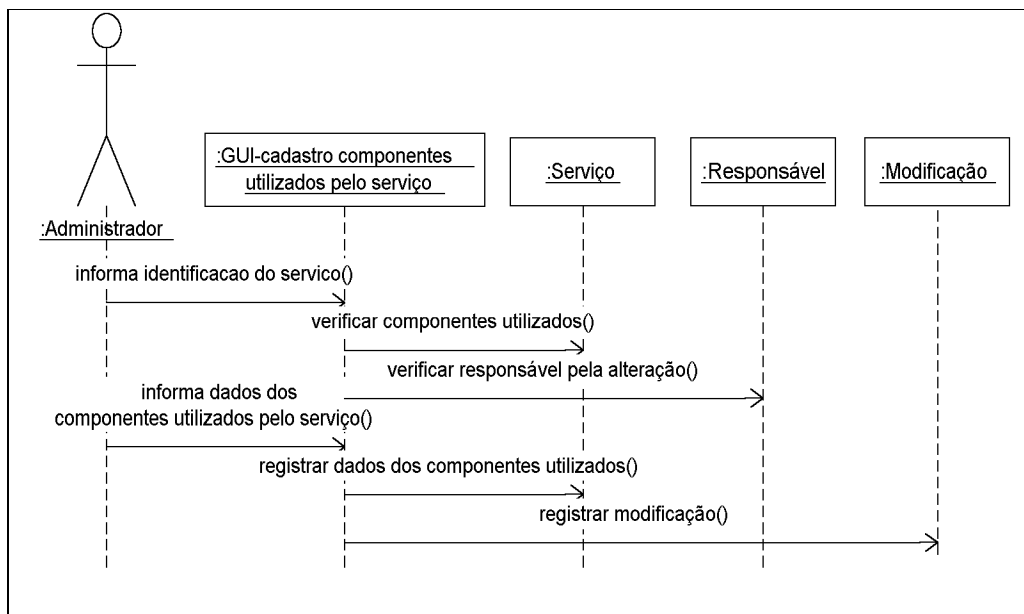


Figura D.6 – Diagrama de Seqüência – Cadastramento dos componentes utilizados pelo serviço.

Caso de Uso: Cadastramento dos componentes utilizados pelo serviço - Exclusão.

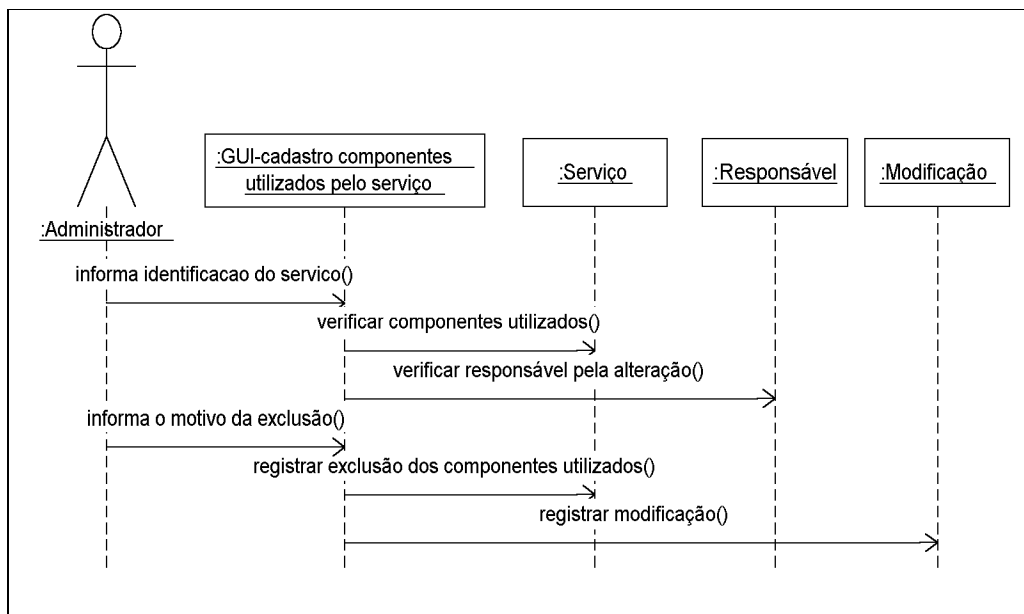


Figura D.7 - Diagrama de Seqüência – Cadastramento dos componentes utilizados pelo serviço – Exclusão.

## e) Caso de Uso: Cadastramento dos problemas com os serviços.

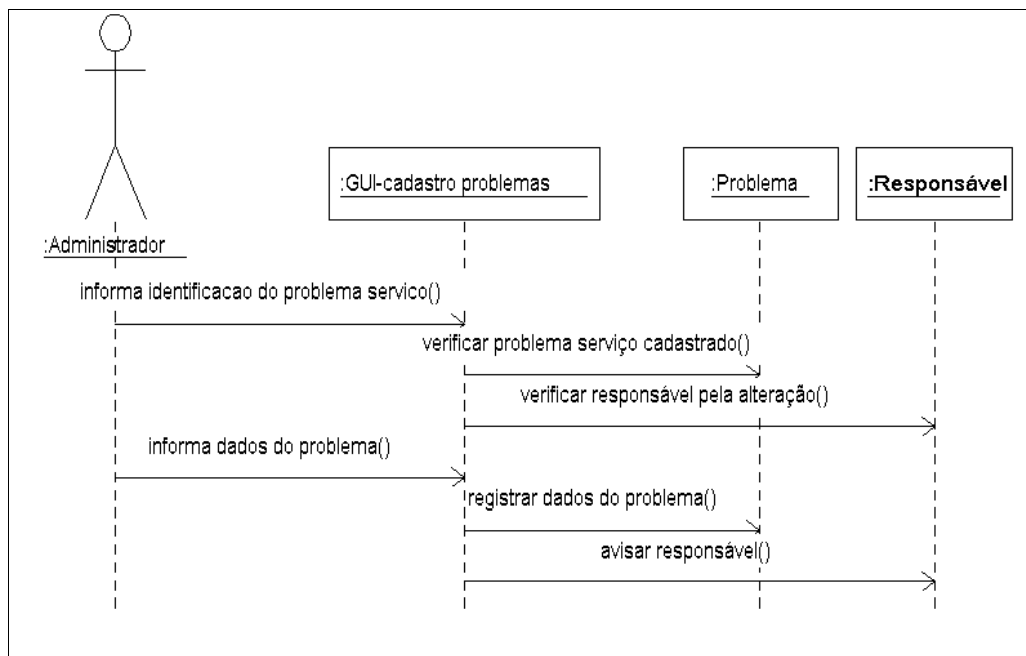


Figura D.8 – Diagrama de Sequência – Cadastramento dos problemas com os serviços.

## Caso de Uso: Cadastramento dos problemas com os serviços - Exclusão.

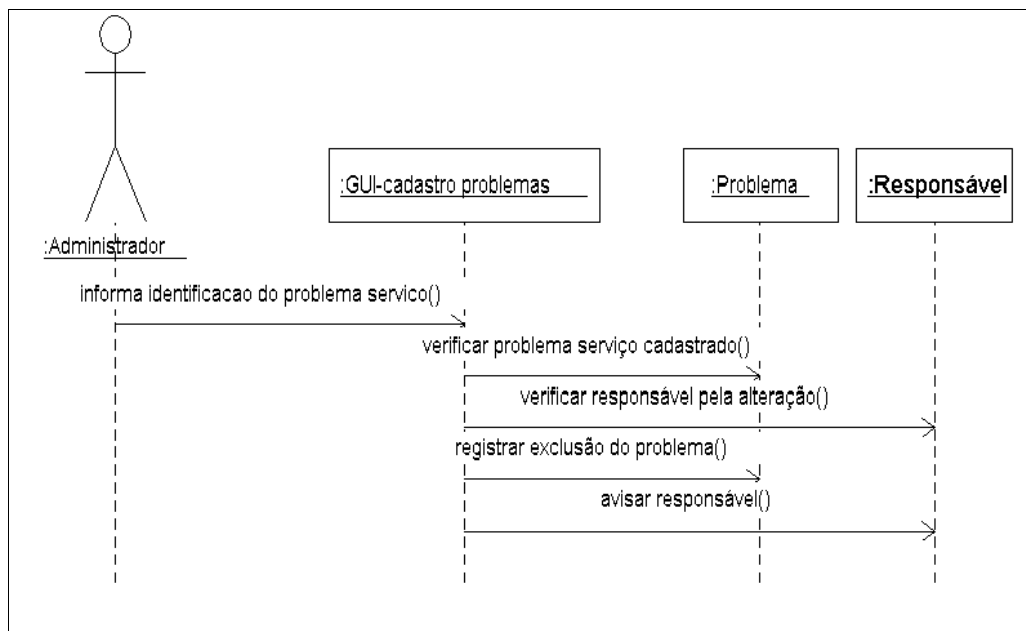


Figura D.9 – Diagrama de Sequência – Cadastramento dos problemas com os serviços – Exclusão.



## f) Caso de Uso: Estorno das modificações.

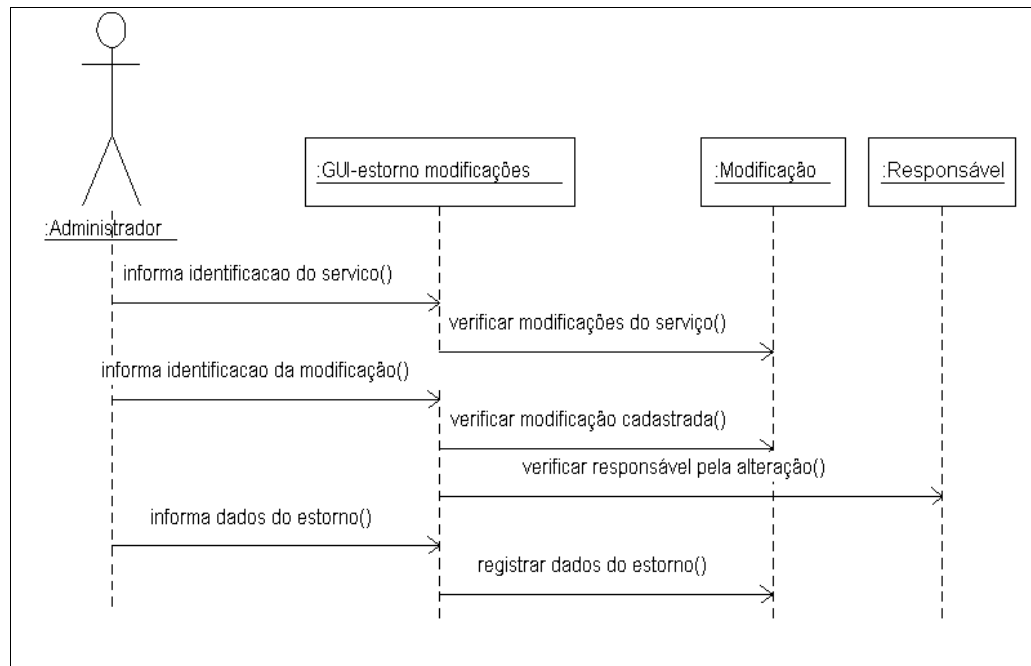


Figura D.10 – Diagrama de Sequência – Estorno das modificações.

## g) Caso de Uso: Alteração do estado do serviço.

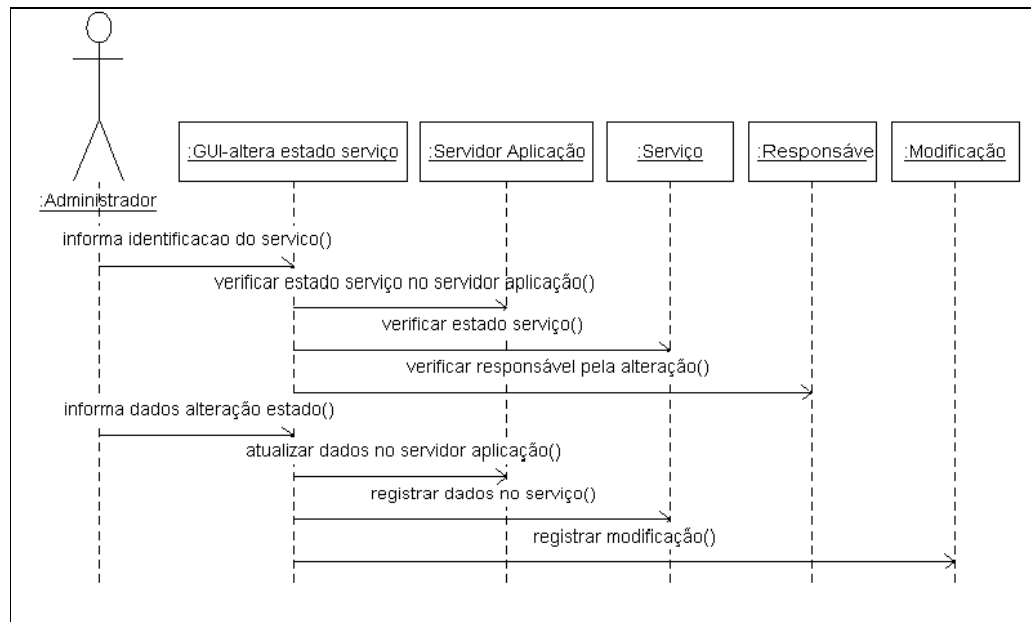


Figura D.11 – Diagrama de Sequência – Alteração do estado do serviço.

## h) Caso de Uso: Cadastramento de Responsáveis.

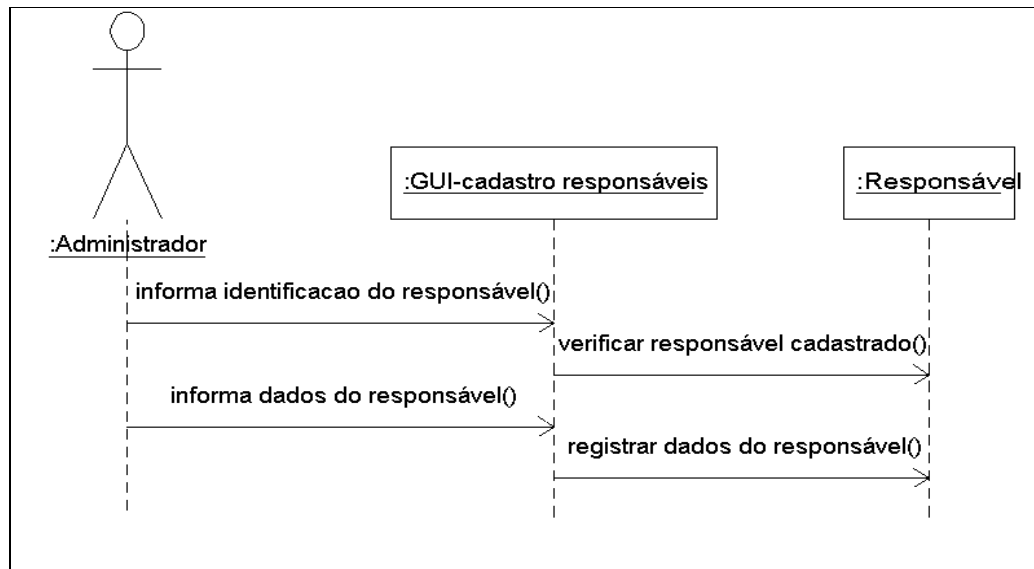


Figura D.12 – Diagrama de Sequência - Cadastramento de Responsáveis.

## Caso de Uso: Cadastramento de Responsáveis - Exclusão.

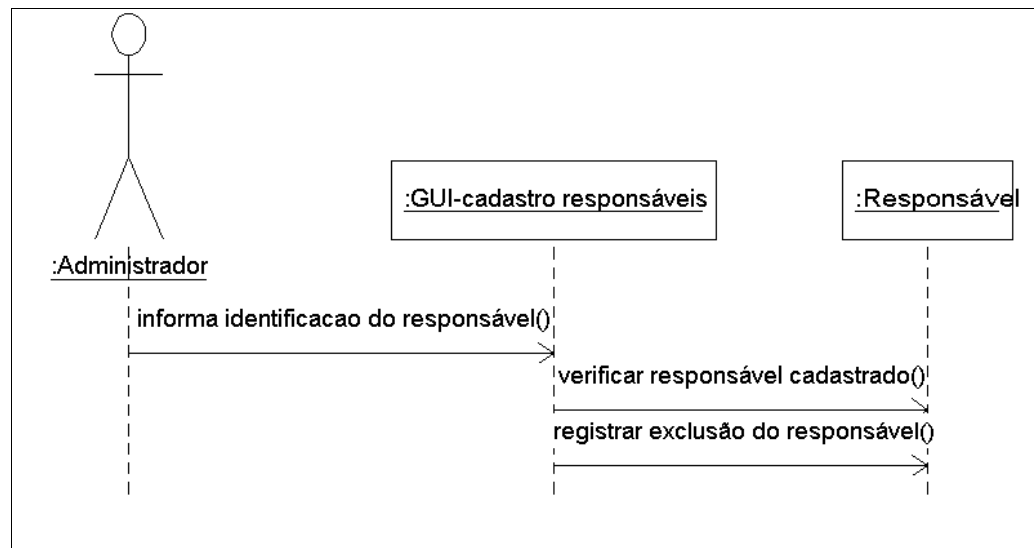


Figura D.13 – Diagrama de Sequência - Cadastramento de Responsáveis – Exclusão.

## 2) Grupo de controle

i) Caso de Uso: Administração da data de expiração dos serviços.

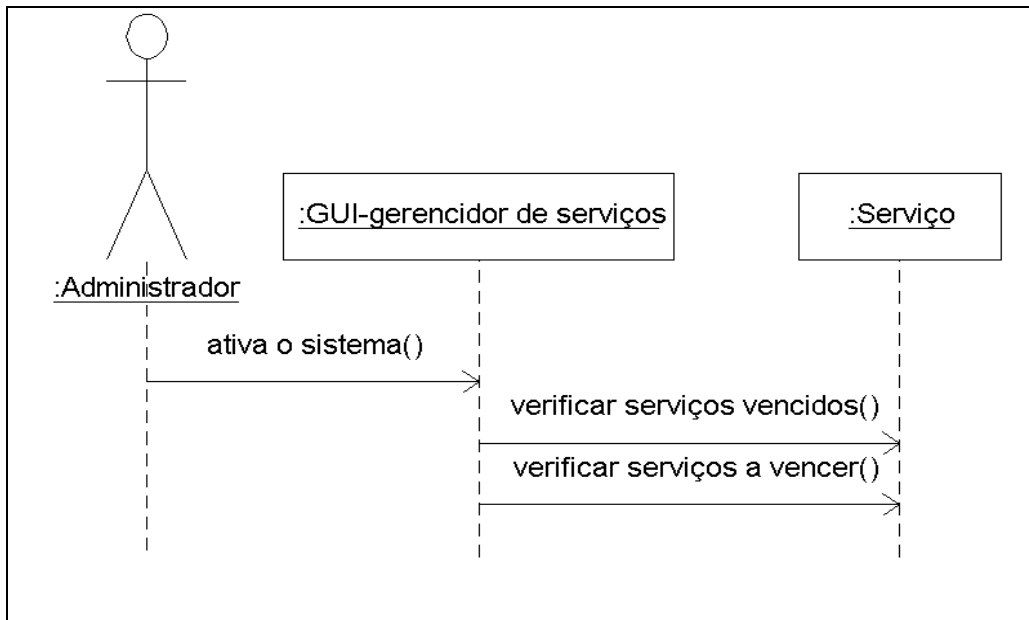


Figura D.14 – Diagrama de Seqüência - Administração da data de expiração dos serviços.

j) Caso de Uso: Administração dos problemas não solucionados relativos aos serviços.

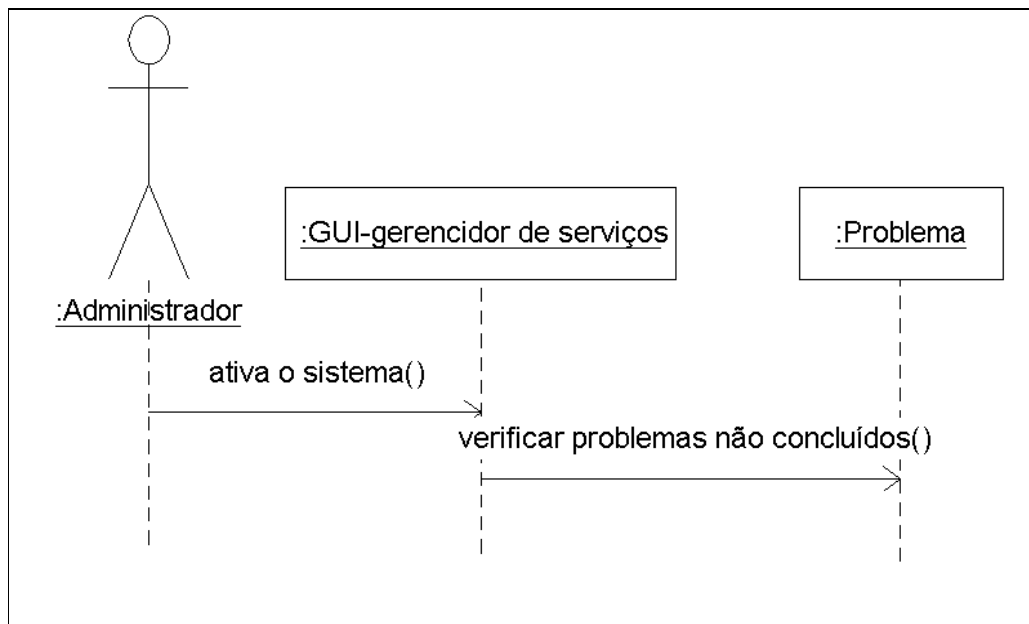


Figura D.15 – Diagrama de Seqüência - Administração dos problemas não solucionados relativos aos serviços.

k) Caso de Uso: Administração do estado dos serviços.

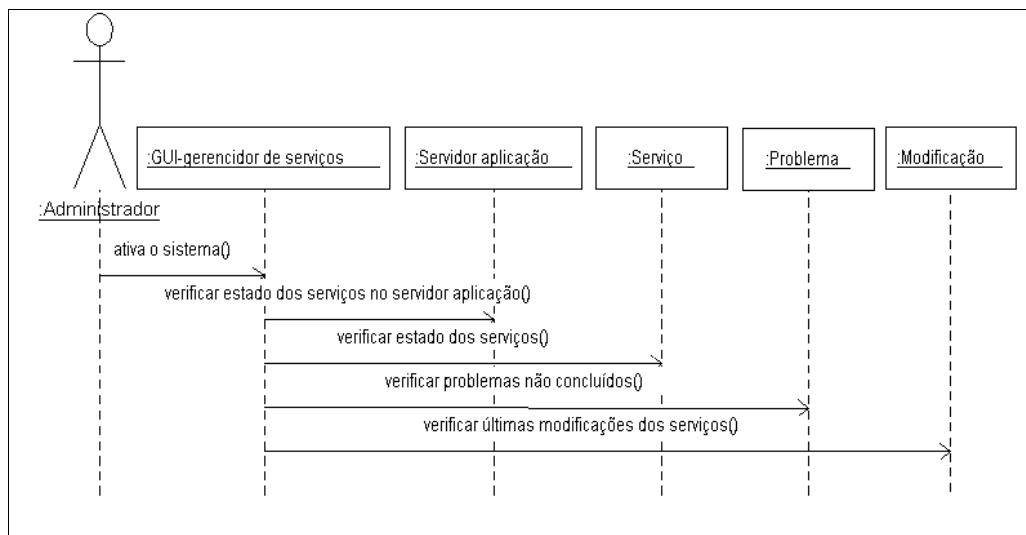


Figura D.16 – Diagrama de Seqüência - Administração do estado dos serviços.

### 3) Grupo das consultas ou relatórios

m) Caso de Uso: Analisa as informações contidas na base de dados através de um visualizador.

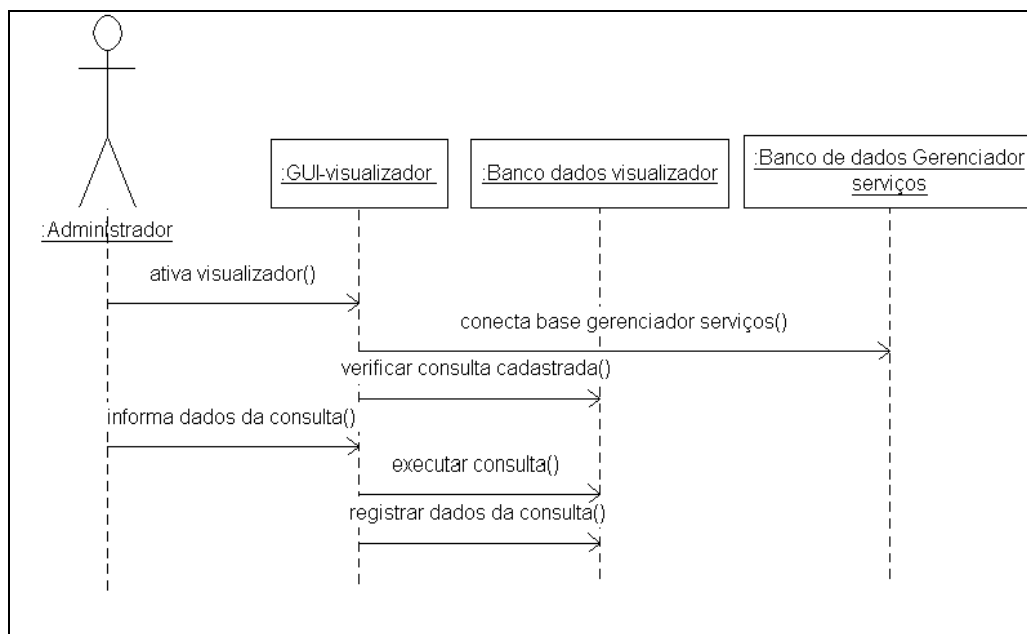


Figura D.17 – Diagrama de Seqüência - Analisa as informações contidas na base de dados através de um visualizador.

## Apêndice E – Protótipo das telas do gerenciador de serviços.

### E.1 Introdução

As informações contidas nos Diagramas de Caso de Uso (Apêndice A), nos Diagramas de Sequência (Apêndice D) e no Diagrama de Classes (item 3.7) viabilizam a confecção dos desenhos das interfaces gráficas - GUI que compõem o protótipo das telas do gerenciador de serviços aqui apresentadas. Esse apêndice, além de destacar as telas do sistema apresenta descrições sobre as ações de comando e os campos que compõem as telas.

### E.2 Tela Principal do sistema

A funcionalidade da tela principal do gerenciador de serviços é atender aos casos de uso de administração da data de expiração dos serviços (Apêndice A-i), da administração dos problemas não solucionados relativos aos serviços (Apêndice A-j) e da administração do estado dos serviços (Apêndice A-k).

**Sistema Gerenciador de Serviços**

Serviços que estão ativos, mas estão com problemas.

| # | Criticidade | Serviço | Estado | Problema | 3 últimas modificações |
|---|-------------|---------|--------|----------|------------------------|
|   |             |         |        |          |                        |
|   |             |         |        |          |                        |
|   |             |         |        |          |                        |
|   |             |         |        |          |                        |
|   |             |         |        |          |                        |
|   |             |         |        |          |                        |
|   |             |         |        |          |                        |
|   |             |         |        |          |                        |

Serviços que estão com problemas não concluídos

| # | Criticidade | Dt Registro | Problema | Serviço | Responsável |
|---|-------------|-------------|----------|---------|-------------|
|   |             |             |          |         |             |
|   |             |             |          |         |             |
|   |             |             |          |         |             |
|   |             |             |          |         |             |
|   |             |             |          |         |             |
|   |             |             |          |         |             |
|   |             |             |          |         |             |
|   |             |             |          |         |             |
|   |             |             |          |         |             |

Data de expiração dos serviços

| # | Dt Contrato | Serviço | Fornecedor | Contrato |
|---|-------------|---------|------------|----------|
|   |             |         |            |          |
|   |             |         |            |          |
|   |             |         |            |          |
|   |             |         |            |          |
|   |             |         |            |          |
|   |             |         |            |          |
|   |             |         |            |          |
|   |             |         |            |          |
|   |             |         |            |          |

Figura E.1 – Protótipo de tela – Tela principal.

Tabela E.1 – Campos da Tela principal.

Campos da tela<sup>5</sup>:

| N.  | Nome  | Valores Válidos            | Tamanho    | Tipo | Restrições  |
|---|---|----------------------------|------------|------|---|
| Serviços que estão ativos, mas estão com problemas. |   |                            |            |      |   |
| 1   | #   | Maior que Zero             | 02 dígitos | N    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 2   | Criticidade                                   | Alta; Média; Baixa.        | Mx. 5      | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 3   | Serviço (Nome + versão)                       | Qq. ASC II                 | Mx. 35     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 4   | Estado  | Ativo; Inativo; Bloqueado. | Mx. 9      | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 5   | Problema                                      | Qq. ASC II                 | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 6   | 3 últimas modificações                        | Qq. ASC II                 | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| Serviços que estão com problemas não concluídos     |   |                            |            |      |   |
| 7   | #   | Maior que Zero             | 02 dígitos | N    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 8   | Criticidade                                   | Alta; Média; Baixa.        | Mx. 5      | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 9   | Dt Registro (Data de registro)                | Maior 01/01/2005           | Mx. 10     | D    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 10  | Problema                                      | Qq. ASC II                 | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 11  | Serviço                                       | Qq. ASC II                 | Mx. 24     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 12  | Responsável                                   | Qq. ASC II                 | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| Data de expiração dos serviços                      |   |                            |            |      |   |
| 13  | #   | Maior que Zero             | 02 dígitos | N    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 14  | Dt. Contrato (Data de vencimento do contrato) | Maior 01/01/2005           | Mx. 10     | D    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 15  | Serviço                                       | Qq. ASC II                 | Mx. 24     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 16  | Fornecedor                                    | Qq. ASC II                 | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |
| 17  | Contrato                                      | Qq. ASC II                 | Mx. 10     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul> |

<sup>5</sup> Legenda coluna: (a) Valores Válidos: Qq = qualquer; (b) Tamanho: Mx. = máximo; (c) Tipo: N = numérico, T = texto, D = data, L = lógico.

### E.3 Tela: Cadastro de Serviços

A funcionalidade da tela do cadastro de serviços atende aos casos de uso do cadastramento e manutenção do acervo de serviços (Apêndice A-a), do cadastramento e administração dos serviços adquiridos de terceiros (Apêndice A-b) e da divulgação automática de informações sobre os serviços (Apêndice A-l).

Figura E.2 – Protótipo de tela – Cadastro de Serviços.

Tabela E.2 – Comandos da Tela do Cadastro de Serviços.

Comandos da tela:

| N. | Nome      | Ação   | Restrições  |
|----|-----------|--|---|
| 1  | Pesquisar | Obter como resultado de pesquisa uma lista de serviços.  | Efetua a pesquisa somente quando um ou mais dos seis campos de pesquisa é preenchido. |
| 2  | Salvar    | Gravar o registro. Durante a ação de salvar ocorre a transferência das informações digitadas na tela para a base de dados. | Habilitado quando os campos obrigatórios estiverem preenchidos.                       |
| 3  | Alterar   | Permitir alteração dos campos apresentados na tela.  | Habilitado após a identificação do serviço.   |
| 4  | Excluir   | Excluir o registro.  | Habilitado após a identificação do serviço.   |

Tabela E.3 – Campos da Tela do Cadastro de serviços.

Campos da tela:

| N.                               | Nome  | Valores Válidos   | Tamanho | Tip | Restrições   |
|----------------------------------|---|---|---------|-----|--|
| Pesquisar e selecionar o serviço |   |   |         |     |  |
| 1                                | Serviço                                     | Qq. ASC II  | Mx. 20  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Lista.</li> <li>▪ Alterável.</li> </ul>     |
| 2                                | Versão                                      | Qq. ASC II  | Mx. 15  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Lista.</li> <li>▪ Alterável.</li> </ul>     |
| 3                                | Projeto                                     | Qq. ASC II  | Mx. 20  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Lista.</li> <li>▪ Alterável.</li> </ul>     |
| 4                                | Tipo serviço                                | Qq. ASC II  | Mx. 20  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Lista.</li> <li>▪ Alterável.</li> </ul>     |
| 5                                | Fornecedor                                  | Qq. ASC II  | Mx. 40  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Lista.</li> <li>▪ Alterável.</li> </ul>     |
| 6                                | Estado                                      | Ativo; Inativo; Bloqueado; Em instalação; Em configuração; Em reconfiguração. | Mx. 9   | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Lista.</li> <li>▪ Alterável.</li> </ul>     |
| Aspectos do serviço              |   |   |         |     |  |
| 7                                | Id. Serviço (Identificador do serviço)      | Maior que zero  | Mx. 8   | N   | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                                      |
| 8                                | Versão                                      | Qq. ASC II  | Mx. 15  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável durante o ato de inclusão.</li> </ul> |
| 9                                | Nome  | Qq. ASC II  | Mx. 20  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                           |
| 10                               | Descrição                                   | Qq. ASC II  | Mx. 50  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                           |
| 11                               | Projeto                                     | Qq. ASC II  | Mx. 20  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                           |
| 12                               | Dt. Homologação (Data de homologação)       | Maior 01/01/2005  | Mx. 10  | D   | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                           |
| 13                               | Resp. Homol. (Responsável pela homologação) | Qq. ASC II  | Mx. 40  | T   | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Alterável</li> </ul>                         |
| 14                               | Dt. Inclusão (Data de inclusão no registro) | Maior 01/01/2005  | Mx. 10  | D   | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                           |



## Continuação dos campos da tela.

|   |   |  |              |   |   |
|---|---|--|--------------|---|---|
| 15  | Dt. Ult. Modif.<br>(Data da última modificação)     | Maior do que<br>Dt. Inclusão.  | Mx.<br>10    | D | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                             |
| 16  | Dt. Desativação<br>(Data de desativação)            | Maior do que<br>Dt. Inclusão.  | Mx.<br>10    | D | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                             |
| 17  | Dt. Expiração<br>(Data de expiração)                | Maior do que<br>Dt. Inclusão.  | Mx.<br>10    | D | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 18  | Tempo fora do ar <sup>6</sup>                       | Maior que<br>Zero.   | 8<br>dígitos | N | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 19  | Plataforma operacional                              | Qq. ASC II   | Mx.<br>60    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 20  | Período de atividade                                | Números e caracteres separadores   | Mx.<br>60    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 21  | Responsável   | Qq. ASC II   | Mx.<br>40    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                  |
| 22  | Tipo serviço  | Qq. ASC II   | Mx.<br>20    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                  |
| 23  | Fornecedor  | Qq. ASC II   | Mx.<br>40    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 24  | Contrato  | Números e caracteres separadores   | Mx.<br>10    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 25  | Dt. Venc. Contrato (Data de vencimento do contrato) | Maior do que<br>Dt. Inclusão   | Mx.<br>10    | D | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 26  | Local instalação<br>(Local de instalação)           | Qq. ASC II   | Mx.<br>40    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                  |
| 27  | Dt. Instalação<br>(Data de instalação)              | Maior do que<br>01/01/2000   | Mx.<br>10    | D | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Alterável.</li> </ul>                  |
| 28  | Criticidade   | Alto, Médio ou Baixo.  | Mx. 5        | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Lista</li> <li>▪ Alterável.</li> </ul> |
| 29  | Linguagem programação                               | Qq. ASC II   | Mx.<br>50    | T | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| 30  | Estado  | Ativo; Inativo; Bloqueado; Em instalação; Em configuração; Em re-configuração. | Mx. 9        | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório.</li> <li>▪ Lista</li> <li>▪ Alterável.</li> </ul> |
| 31  | Configuração<br>(do serviço e do ambiente)          | Qq. ASC II   | Mx.<br>8000  | T | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório.</li> <li>▪ Alterável.</li> </ul>              |
| Interface do serviço – ver item E.4.                |   |  |              |   |   |
| Componentes utilizados pelo serviço – ver item E.5. |   |  |              |   |   |

<sup>6</sup> Tempo previsto em minutos que o serviço pode ficar fora do ar durante um dia.

#### E.4 Tela: Cadastro da interface do serviço

A funcionalidade da tela do cadastro da interface do serviço atende aos casos de uso de cadastramento e manutenção das informações sobre a interface dos serviços (Apêndice A-c) e da divulgação automática de informações sobre os serviços (Apêndice A-l).

Figura E.3 – Protótipo de tela – Cadastro de Interface de um serviço.

Tabela E.4 – Comandos da Tela do Cadastro da interface do serviço.

Comandos da tela:

| N. | Nome    | Ação  | Restrições  |
|----|---------|---|---|
| 1  | Salvar  | Gravar os registros que compõem a interface do serviço. Durante a ação de salvar ocorre a transferência das informações digitadas na tela para a base de dados. | Habilitado quando os campos obrigatórios estiverem preenchidos. |
| 2  | Alterar | Permitir alteração dos campos apresentados na tela.   | Habilitado após a identificação do serviço.                     |
| 3  | Excluir | Excluir os registros que compõem a interface.   | Habilitado após a identificação do serviço.                     |

Tabela E.5 – Campos da Tela do Cadastro da interface do serviço.

Campos da tela:

| N.        | Nome    | Valores Válidos | Tamanho | Tipo | Restrições  |
|-----------|---------|-----------------|---------|------|---|
| Selecione |         |                 |         |      |   |
| 1         | Serviço | Qq. ASC II      | Mx. 20  | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 2         | Versão  | Qq. ASC II      | Mx. 15  | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |

Continuação dos campos da tela.

| Interface |                |                          |            |   |   |
|-----------|----------------|--------------------------|------------|---|---|
| 3         | #              | Maior que Zero           | 05 dígitos | N | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| 4         | Nome argumento | Qq. ASC II               | Mx. 30     | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 5         | Descrição      | Qq. ASC II               | Mx. 50     | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 6         | Tipo           | Qq. ASC II               | Mx. 20     | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 7         | Tamanho        | Maior que Zero           | 05 dígitos | N | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 8         | Direção        | <i>In, out ou inout.</i> | Mx. 5      | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 9         | Observação     | Qq. ASC II               | Mx. 8000   | T | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Alterável</li> </ul>              |

### E.5 Tela: Cadastro dos componentes utilizados pelo serviço

A funcionalidade da tela do cadastro dos componentes utilizados pelo serviço atende aos casos de uso de cadastramento e manutenção dos componentes utilizados pelos serviços (Apêndice A-d) e à divulgação automática de informações sobre os serviços (Apêndice A-l).

Figura E.4 – Protótipo de tela- Cadastro dos componentes utilizados pelo serviço.

Tabela E.6 – Comandos da Tela do Cadastro dos componentes utilizados pelo serviço.  
Comandos da tela:

| N. | Nome    | Ação  | Restrições  |
|----|---------|---|---|
| 1  | Salvar  | Gravar os registros que compõem os componentes utilizados pelo serviço. Durante a ação de salvar ocorre a transferência das informações digitadas na tela para a base de dados. | Habilitado quando os campos obrigatórios estiverem preenchidos. |
| 2  | Alterar | Permitir alteração dos campos apresentados na tela.   | Habilitado após a identificação do serviço.                     |
| 3  | Excluir | Excluir os registros que compõem os componentes utilizados pelo serviço.  | Habilitado após a identificação do serviço.                     |

Tabela E.7 – Campos da Tela do Cadastro dos componentes utilizados pelo serviço.  
Campos da tela:

| N.                                  | Nome  | Valores Válidos    | Tamanho    | Ti po | Restrições  |
|-------------------------------------|---|--------------------|------------|-------|---|
| Selecione                           |   |                    |            |       |   |
| 1                                   | Serviço   | Qq. ASC II         | Mx. 20     | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 2                                   | Versão  | Qq. ASC II         | Mx. 15     | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| Componentes utilizados pelo serviço |   |                    |            |       |   |
| 3                                   | #   | Maior que Zero     | 02 dígitos | N     | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| 4                                   | Id. Componente (Identificador do componente utilizado pelo serviço) | Maior que Zero     | 08 dígitos | N     | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| 5                                   | Versão  | Qq. ASC II         | Mx. 15     | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 6                                   | Descrição   | Qq. ASC II         | Mx. 50     | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 7                                   | Local Instalação  | Qq. ASC II         | Mx. 40     | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 8                                   | Observação  | Qq. ASC II         | Mx. 8000   | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Alterável</li> </ul>              |
| 9                                   | Estado  | Ativo ou Inativo   | Mx. 7      | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 10                                  | Dt. Registro (Data de registro)                                     | Data e hora atuais | Mx. 10     | D     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 11                                  | Responsável   | Qq. ASC II         | Mx. 40     | T     | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |

## E.6 Tela: Cadastro dos problemas com o serviço

A funcionalidade da tela do cadastro de problemas atende ao caso de uso de cadastramento e manutenção dos problemas e soluções que ocorrem com os serviços (Apêndice A-e).

Figura E.5 – Protótipo de tela – Cadastro de Problemas.

Tabela E.8 – Comandos da Tela do Cadastro de Problemas.

Comandos da tela:

| N. | Nome    | Ação   | Restrições  |
|----|---------|--|---|
| 1  | Salvar  | Gravar o registro. Durante a ação de salvar ocorre a transferência das informações digitadas na tela para a base de dados. | Habilitado quando os campos obrigatórios estiverem preenchidos. |
| 2  | Alterar | Permitir alteração dos campos apresentados na tela.  | Habilitado após a identificação do problema.                    |
| 3  | Excluir | Excluir o registro.  | Habilitado após a identificação do problema.                    |

Tabela E.9 – Campos da Tela do Cadastro de Problemas.  
Campos da tela:

| N.        | Nome                             | Valores Válidos                     | Tamanho    | Tipo | Restrições  |
|-----------|----------------------------------|-------------------------------------|------------|------|---|
| Selecione |                                  |                                     |            |      |   |
| 1         | Serviço                          | Qq. ASC II                          | Mx. 20     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 2         | Versão                           | Qq. ASC II                          | Mx. 15     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 3         | Data registro Problema           | Data e hora atuais                  | Mx. 10     | D    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| Problema  |                                  |                                     |            |      |   |
| 4         | Situação                         | Em aberto, Em andamento, Concluído. | Mx. 12     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 5         | Data da constatação do problema  | Maior que 01/01/2005                | Mx. 10     | D    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 6         | Criticidade                      | Alta, Média ou Baixa.               | Mx. 5      | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 7         | Tipo                             | Qq. ASC II                          | Mx. 20     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 8         | Responsável                      | Qq. ASC II                          | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 9         | Descrição                        | Qq. ASC II                          | Mx. 4000   | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| Solução   |                                  |                                     |            |      |   |
| 10        | Data de encerramento do problema | Maior que data constatação          | Mx. 10     | D    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 11        | Tempo fora do ar <sup>7</sup>    | Maior que Zero                      | 08 dígitos | N    | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Alterável</li> </ul>              |
| 12        | Responsável                      | Qq. ASC II                          | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 13        | Descrição                        | Qq. ASC II                          | Mx. 4000   | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |

### E.7 Tela: Estorno das últimas modificações

A funcionalidade da tela do estorno das últimas modificações atende ao caso de uso do anulamento (ou Estorno) das últimas modificações (nos componentes utilizados, na interface e nas informações dos serviços) que ocorreram com os serviços (Apêndice A-f).

<sup>7</sup> O tempo fora do ar é mensurado em minutos.

Figura E.6 – Protótipo de tela – Estorno das últimas modificações.

Tabela E.10 – Comandos da Tela do Estorno das últimas modificações.

Comandos da tela:

| N. | Nome      | Ação  | Restrições  |
|----|-----------|---|---|
| 1  | Pesquisar | Obter como resultado de pesquisa uma lista com as últimas modificações dos serviços.  | Efetua a pesquisa quando a data início e término são preenchidos. |
| 2  | Salvar    | Gravar o registro de estorno. Durante a ação de salvar ocorre a transferência das informações digitadas na tela para a base de dados. | Habilitado quando os campos obrigatórios estiverem preenchidos.   |

Tabela E.11 – Campos da Tela do Estorno das últimas modificações.

Campos da tela:

| N.                                   | Nome         | Valores Válidos          | Tamanho | Tipo | Restrições   |
|--------------------------------------|--------------|--------------------------|---------|------|--|
| Selecione o período das modificações |              |                          |         |      |  |
| 1                                    | Data início  | Maior do que 01/01/2005  | Mx. 10  | D    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul> |
| 2                                    | Data término | Maior do que data início | Mx. 10  | D    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul> |

Continuação dos campos da tela.

| Selecione a modificação a estornar |  |                           |            |   |  |
|------------------------------------|--|---------------------------|------------|---|--|
| 3                                  | #  | Maior que Zero            | 02 dígitos | N | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 4                                  | Dt. Modif. (Data da modificação)         | Maior que data início     | Mx. 10     | D | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 5                                  | Serviço                                  | Qq. ASC II                | Mx. 20     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 6                                  | Versão                                   | Qq. ASC II                | Mx. 15     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 7                                  | Tipo Modif. (Tipo de modificação)        | Qq. ASC II                | Mx. 20     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 8                                  | Descrição Modificação                    | Qq. ASC II                | Mx. 40     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 9                                  | Desc. Serviço (Descrição do serviço)     | Qq. ASC II                | Mx. 50     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 10                                 | Responsável Modificação                  | Qq. ASC II                | Mx. 40     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| Estorno                            |  |                           |            |   |  |
| 11                                 | Serviço                                  | Qq. ASC II                | 08 dígitos | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 12                                 | Versão                                   | Qq. ASC II                | Mx. 15     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 13                                 | Data modificação                         | Maior que data início     | Mx. 10     | D | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 14                                 | Responsável                              | Qq. ASC II                | Mx. 40     | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>          |
| 15                                 | Data estorno                             | Maior que Dt. Modificação | Mx. 10     | D | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul> |
| 16                                 | Descrição de como desfazer a modificação | Qq. ASC II                | Mx. 4000   | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul> |

## E.8 Tela: Altera Estado do Serviço

A funcionalidade da tela que altera o estado do serviço atende ao caso de uso de alteração do estado do serviço (Apêndice A-g).



Figura E.7 – Protótipo de tela – Altera Estado do Serviço.

Tabela E.12 – Comandos da Tela - Altera Estado do Serviço.

Comandos da tela:

| N. | Nome   | Ação   | Restrições  |
|----|--------|--|---|
| 1  | Salvar | Gravar o registro da alteração do estado do serviço. Durante a ação de salvar ocorre a transferência das informações digitadas na tela para a base de dados. | Habilitado quando os campos obrigatórios estiverem preenchidos. |

Tabela E.13 – Campos da Tela - Altera Estado do Serviço.

Campos da tela:

| N.                  | Nome    | Valores Válidos | Tamanho    | Tipo | Restrições  |
|---------------------|---------|-----------------|------------|------|---|
| Selecione           |         |                 |            |      |   |
| 1                   | Projeto | Qq. ASC II      | Mx. 20     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 2                   | Serviço | Qq. ASC II      | Mx. 20     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 3                   | Versão  | Qq. ASC II      | Mx. 15     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| Selecione o serviço |         |                 |            |      |   |
| 4                   | #       | Maior que Zero  | 02 dígitos | N    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                               |
| 5                   | Serviço | Qq. ASC II      | Mx. 20     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                               |
| 6                   | Versão  | Qq. ASC II      | Mx. 15     | T    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                               |

Continuação dos campos da tela.

|                             |  |                            |        |   |   |
|-----------------------------|--|----------------------------|--------|---|---|
| 7                           | Descrição  | Qq. ASC II                 | Mx. 50 | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| 8                           | Estado   | Ativo, Inativo, Bloqueado. | Mx. 9  | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| Alterar o Estado do serviço |  |                            |        |   |   |
| 9                           | Estado atual – Cad. Serviços (Estado atual – cadastro de serviços) | Ativo, Inativo, Bloqueado. | Mx. 9  | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| 10                          | Estado atual – Servidor Aplicação                                  | Ativo, Inativo, Bloqueado. | Mx. 9  | T | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| 11                          | NOVO estado  | Ativo, Inativo, Bloqueado. | Mx. 9  | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 12                          | Solicitante  | Qq. ASC II                 | Mx. 40 | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |
| 13                          | Motivo da alteração  | Qq. ASC II                 | Mx. 40 | T | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |

### E.9 Tela: Cadastro dos Responsáveis

A funcionalidade da tela do cadastro de responsáveis atende ao caso de uso de cadastramento dos usuários responsáveis pelos serviços, pela homologação, pelos registros de modificações, pelos registros de problemas e soluções, pelos registros dos componentes utilizados pelos serviços e pesquisas no acervo (Apêndice A-h).

**Sistema Gerenciador de Serviços - Cadastro de Responsáveis**

Responsável

Identificador:

Nome:

E-mail:

Telefone:

Tipo de Responsabilidade:

Enviar e-mail informando sobre as modificações e inclusões dos serviços:

Figura E.8 – Protótipo de tela – Cadastro dos Responsáveis.

Tabela E.14 – Comandos da Tela do Cadastro dos Responsáveis.

Comandos da tela:

| N. | Nome    | Ação   | Restrições  |
|----|---------|--|---|
| 1  | Salvar  | Gravar o registro. Durante a ação de salvar ocorre a transferência das informações digitadas na tela para a base de dados. | Habilitado quando os campos obrigatórios estiverem preenchidos. |
| 2  | Alterar | Permitir alteração dos campos apresentados na tela.  | Habilitado após a identificação do responsável.                 |
| 3  | Excluir | Excluir o registro.  | Habilitado após a identificação do responsável.                 |

Tabela E.15 – Campos da Tela do Cadastro dos Responsáveis.

Campos da tela:

| N.          | Nome  | Valores Válidos | Tamanho    | Tipo | Restrições  |
|-------------|---|-----------------|------------|------|---|
| Responsável |   |                 |            |      |   |
| 1           | Identificador   | Maior que Zero  | 08 dígitos | N    | <ul style="list-style-type: none"> <li>▪ Apresentação</li> <li>▪ Não alterável</li> </ul>                           |
| 2           | Nome  | Qq. ASC II      | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 3           | E-mail  | Qq. ASC II      | Mx. 40     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 4           | Telefone  | Qq. ASC II      | Mx. 15     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento não obrigatório</li> <li>▪ Alterável</li> </ul>              |
| 5           | Tipo de Responsabilidade  | Qq. ASC II      | Mx. 30     | T    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Alterável</li> </ul>                  |
| 6           | Enviar e-mail informando sobre as modificações e inclusões dos serviços | Sim ou Não      |            | L    | <ul style="list-style-type: none"> <li>▪ Preenchimento obrigatório</li> <li>▪ Lista</li> <li>▪ Alterável</li> </ul> |

## E.10 Conclusão

Entre todos os casos de uso analisados no apêndice A o único que não está contemplado neste apêndice E é o caso de uso do apêndice A-m “Disponibilização da base de informações do gerenciador de serviços”, pois depende da ferramenta externa vizualizadora que será eleita para elaborar as diversas consultas e relatórios necessários para atender as ações operacionais e gerenciais do sistema.

A prototipação da interface homem-computador (IHC) é uma etapa muito importante para o refino dos requisitos e funcionalidades, sendo assim, através da construção destas telas é possível identificar, no domínio dos requisitos, a necessidade de novos atributos e a remoção de atributos redundantes. Ações similares também ocorrem no domínio das funcionalidades. Além dos atributos e funcionalidades do domínio do negócio é possível identificar e destacar as classes que trabalham com as interfaces gráficas.

## Apêndice F –Projeto dos Testes de Integração – Contratos de Classes.

As especificações detalhadas dos contratos das classes são fundamentais para a elaboração dos casos de teste. Estas especificações são organizadas e apresentadas na forma de tabelas (F.1 até F.11) que contêm, de forma padronizada, as informações pertinentes à classe, ao método que disponibiliza, à estrutura interna utilizada e às condições, exceções e saídas tratadas (PRESSMAN, 2001).

F.1 Tabela das características do contrato de classe desenvolvidos para a operação **registrar dados do serviço()** da classe Serviço.

Tabela F.1 – Contrato da classe Serviço, operação: registrar dados do serviço().

|                          |   |
|--------------------------|---|
| <b>Classe</b>            | Serviço.  |
| <b>Nome operação</b>     | registrar dados do serviço()  |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_registraSrv passando como parâmetros as informações pertinentes a um serviço.  |
| <b>Tipo</b>              | Operação de classe.   |
| <b>Referência</b>        | Casos de uso: (Apêndice A-a) cadastramento e manutenção do acervo de serviços, (Apêndice A-b) cadastramento e administração dos serviços adquiridos de terceiros e (Apêndice A-d) cadastramento e manutenção dos componentes do serviço.                          |
| <b>Observações</b>       | Os atributos desta classe contemplam informações de aspectos, operações, configurações e responsabilidades. O registro destas informações é o alicerce para a gestão dos serviços.  |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.  |
| <b>Saída</b>             |   |
| <b>Pré-condições</b>     | É necessária a existência cadastral dos usuários responsáveis por homologar e desenvolver o serviço.  |
| <b>Pós-condições</b>     | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – registrou as informações com sucesso, era uma inclusão.<br>1 – registrou as informações com sucesso, era uma alteração.<br>10 – ocorreu erro de validação nas informações a registrar. |

F.2 Tabela das características do contrato de classe desenvolvidos para a operação **verificar serviços vencidos()** da classe Serviço.

Tabela F.2 – Contrato da classe Serviço, operação: verificar serviços vencidos().

|                          |   |
|--------------------------|---|
| <b>Classe</b>            | Serviço.  |
| <b>Nome operação</b>     | verificar serviços vencidos()   |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_verificaSrvVencidos passando como parâmetros as informações pertinentes a um período de tempo. |
| <b>Tipo</b>              | Operação de classe.   |
| <b>Referência</b>        | Caso de uso: (Apêndice A-i) Administração da data de expiração dos serviços.  |
| <b>Observações</b>       | O próprio sistema informa o administrador dos contratos de licenciamento dos serviços que estão na situação de vencidos.                  |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.  |
| <b>Saída</b>             | Como resultado da execução da operação é gerada uma lista ordenada por data que contém os serviços vencidos.                              |

Continuação das características do contrato.

|                      |   |
|----------------------|---|
| <b>Pré-condições</b> |   |
| <b>Pós-condições</b> | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – existem serviços vencidos.<br>1 – não existem serviços vencidos. |

F.3 Tabela das características do contrato de classe desenvolvidos para a operação **verificar serviços a vencer()** da classe Serviço.

Tabela F.3 – Contrato da classe Serviço, operação: verificar serviços a vencer().

|                          |   |
|--------------------------|---|
| <b>Classe</b>            | Serviço.  |
| <b>Nome operação</b>     | verificar serviços a vencer()   |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_verificaSrvAvencer passando como parâmetros as informações pertinentes a um período de tempo.    |
| <b>Tipo</b>              | Operação de classe.   |
| <b>Referência</b>        | Caso de uso: (Apêndice A-i) Administração da data de expiração dos serviços.  |
| <b>Observações</b>       | O próprio sistema informa o administrador dos contratos de licenciamento dos serviços que estão na situação a vencer.                       |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.  |
| <b>Saída</b>             | Como resultado da execução da operação é gerada uma lista ordenada por data que contém os serviços a vencer.                                |
| <b>Pré-condições</b>     |   |
| <b>Pós-condições</b>     | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – existem serviços a vencer.<br>1 – não existem serviços a vencer. |

F.4 Tabela das características do contrato de classe desenvolvidos para a operação **registrar dados da interface()** da classe Interface.

Tabela F.4 – Contrato da classe Interface, operação: registrar dados da interface().

|                          |  |
|--------------------------|--|
| <b>Classe</b>            | Interface.   |
| <b>Nome operação</b>     | registrar dados da interface()   |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_registraInterface passando como parâmetros as informações pertinentes a um serviço e sua respectiva interface.  |
| <b>Tipo</b>              | Operação de classe.  |
| <b>Referência</b>        | Caso de uso: (Apêndice A-c) cadastramento e manutenção das informações sobre a interface dos serviços.   |
| <b>Observações</b>       |  |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.   |
| <b>Saída</b>             |  |
| <b>Pré-condições</b>     | O cadastramento da interface exige um cadastramento simultâneo com o serviço ou que o serviço já esteja cadastrado.  |
| <b>Pós-condições</b>     | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – registrou as informações com sucesso, era uma inclusão.<br>1 – registrou as informações com sucesso, era uma alteração.<br>10 – ocorreu erro de validação nas informações a registrar.<br>20 – não existe o cadastramento do serviço. |

F.5 Tabela das características do contrato de classe desenvolvidos para a operação **registrar dados do serviço()** da classe Serviço.

Tabela F.5 – Contrato da classe Serviço, operação: registrar dados do serviço().

|                          |  |
|--------------------------|--|
| <b>Classe</b>            | Serviço.   |
| <b>Nome operação</b>     | registrar dados do serviço()   |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_registraSrv passando como parâmetros as informações pertinentes a um serviço e os respectivos componentes utilizados pelo mesmo.  |
| <b>Tipo</b>              | Operação de classe.  |
| <b>Referência</b>        | Caso de uso: (Apêndice A-d) cadastramento e manutenção dos componentes utilizados pelos serviços.  |
| <b>Observações</b>       |  |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.   |
| <b>Saída</b>             |  |
| <b>Pré-condições</b>     | O cadastramento dos componentes utilizados exige um cadastramento simultâneo com o serviço ou que o serviço já esteja cadastrado.  |
| <b>Pós-condições</b>     | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – registrou as informações com sucesso, era uma inclusão.<br>1 – registrou as informações com sucesso, era uma alteração.<br>10 – ocorreu erro de validação nas informações a registrar.<br>20 – não existe o cadastramento do serviço. |

F.6 Tabela das características do contrato de classe desenvolvidos para a operação **registrar modificação()** da classe Modificação.

Tabela F.6 – Contrato da classe Modificação, operação: registrar modificação().

|                          |   |
|--------------------------|---|
| <b>Classe</b>            | Modificação.  |
| <b>Nome operação</b>     | registrar modificação()   |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_registraModificacao passando como parâmetros as informações pertinentes a um serviço e sua respectiva modificação.   |
| <b>Tipo</b>              | Operação de classe.   |
| <b>Referência</b>        | Casos de uso: (Apêndice A-a) cadastramento de serviços; (Apêndice A-b) cadastramento de serviços de terceiros; (Apêndice A-c) cadastramento da interface do serviço; (Apêndice A-d) cadastramento dos componentes utilizados pelo serviço; (Apêndice A-g) alteração do estado do serviço. |
| <b>Observações</b>       | A classificação cronológica dos registros das modificações do serviço detalha tudo o que ocorreu com o serviço e o que está ocorrendo.  |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.  |
| <b>Saída</b>             |   |
| <b>Pré-condições</b>     | O cadastramento da modificação exige um cadastramento simultâneo com o serviço ou que o serviço já esteja cadastrado.   |

Continuação das características do contrato.

|                      |  |
|----------------------|--|
| <b>Pós-condições</b> | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – registrou as informações com sucesso, era uma inclusão.<br>1 – registrou as informações com sucesso, era uma alteração.<br>10 – ocorreu erro de validação nas informações a registrar.<br>20 – não existe o cadastramento do serviço. |
|----------------------|--|

F.7 Tabela das características do contrato de classe desenvolvidos para a operação **registrar dados do problema()** da classe Problema.

Tabela F.7 – Contrato da classe Problema, operação: registrar dados do problema().

|                          |  |
|--------------------------|--|
| <b>Classe</b>            | Problema.  |
| <b>Nome operação</b>     | registrar dados do problema()  |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_registraProblema passando como parâmetros as informações pertinentes a um serviço e seu respectivo problema e/ou solução.   |
| <b>Tipo</b>              | Operação de classe.  |
| <b>Referência</b>        | Caso de uso: (Apêndice A-e) cadastramento e manutenção dos problemas e soluções que ocorrem com os serviços.   |
| <b>Observações</b>       |  |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.   |
| <b>Saída</b>             |  |
| <b>Pré-condições</b>     | O cadastramento do problema exige um cadastramento simultâneo com o serviço ou que o serviço já esteja cadastrado.<br>O registro da solução exige um cadastramento simultâneo do problema ou que o problema já esteja cadastrado.  |
| <b>Pós-condições</b>     | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – registrou as informações com sucesso, era uma inclusão.<br>1 – registrou as informações com sucesso, era uma alteração.<br>10 – ocorreu erro de validação nas informações a registrar.<br>20 – não existe o cadastramento do serviço.<br>30 – não existe o cadastramento do problema. |

F.8 Tabela das características do contrato de classe desenvolvidos para a operação **verificar problemas não concluídos()** da classe Problema.

Tabela F.8 – Contrato da classe Problema, operação: verificar problemas não concluídos().

|                          |   |
|--------------------------|---|
| <b>Classe</b>            | Problema.   |
| <b>Nome operação</b>     | verificar problemas não concluídos()  |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_verificaPrbNaoConcluido passando como parâmetros as informações pertinentes a um período de tempo. |
| <b>Tipo</b>              | Operação de classe.   |
| <b>Referência</b>        | Caso de uso: (Apêndice A-j) Administração dos problemas não solucionados relativos aos serviços.  |
| <b>Observações</b>       | Este recurso permite que o administrador execute a gestão sobre os problemas pendentes.   |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.  |

Continuação das características do contrato.

|                      |   |
|----------------------|---|
| <b>Saída</b>         | Como resultado da execução da operação é gerada uma lista ordenada por criticidade e data que contém os problemas não concluídos.               |
| <b>Pré-condições</b> |   |
| <b>Pós-condições</b> | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – existem problemas pendentes.<br>1 – não existem problemas pendentes. |

F.9 Tabela das características do contrato de classe desenvolvidos para a operação **registrar dados do responsável()** da classe Responsável.

Tabela F.9 – Contrato da classe Responsável, operação: registrar dados do responsável().

|                          |   |
|--------------------------|---|
| <b>Classe</b>            | Responsável.  |
| <b>Nome operação</b>     | registrar dados do responsável()  |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_registraResponsavel passando como parâmetros as informações pertinentes ao usuário responsável.  |
| <b>Tipo</b>              | Operação de classe.   |
| <b>Referência</b>        | Caso de uso: (Apêndice A-h) Cadastramento dos usuários responsáveis pelos serviços, pela homologação, pelos registros de modificações, pelos registros de problemas e soluções, pelos registros dos componentes utilizados pelos serviços e pesquisas no acervo.                      |
| <b>Observações</b>       | Os objetos persistidos desta classe quando associados a serviços detêm o registro do e-mail que viabiliza as ações de acionamento através de mensagens eletrônicas.<br>O registro do e-mail também viabiliza a divulgação dos serviços para todos os usuários cadastrados no sistema. |
| <b>Exceções</b>          | Falha na conexão com o banco de dados.  |
| <b>Saída</b>             |   |
| <b>Pré-condições</b>     |   |
| <b>Pós-condições</b>     | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – registrou as informações com sucesso, era uma inclusão.<br>1 – registrou as informações com sucesso, era uma alteração.<br>10 – ocorreu erro de validação nas informações a registrar.                     |

F.10 Tabela das características do contrato de classe desenvolvidos para a operação **avisar responsável()** da classe Responsável.

Tabela F.10 – Contrato da classe Responsável, operação: avisar responsável().

|                          |  |
|--------------------------|--|
| <b>Classe</b>            | Responsável.   |
| <b>Nome operação</b>     | avisar responsável()   |
| <b>Responsabilidades</b> | Faz a chamada a <i>stored procedure</i> sp_verificaResponsavel para obter o endereço eletrônico do responsável. Após obter o endereço eletrônico o método em questão formata e envia o e-mail através do protocolo SMTP ( <i>Simple Mail Transport Protocol</i> ). |
| <b>Tipo</b>              | Operação de classe.  |



Continuação das características do contrato.

|                      |   |
|----------------------|---|
| <b>Referência</b>    | Casos de uso: (Apêndice A-a) cadastramento de serviços; (Apêndice A-b) cadastramento de serviços de terceiros; (Apêndice A-c) cadastramento da interface do serviço; (Apêndice A-d) cadastramento dos componentes utilizados pelo serviço; (Apêndice A-e) cadastramento dos problemas com os serviços; (Apêndice A-f) estorno das modificações. |
| <b>Observações</b>   | Um dos objetivos deste método é manter os usuários responsáveis (desenvolvedores) atualizados com o acervo e com as ocorrências dos serviços.   |
| <b>Exceções</b>      | Falha na conexão com o banco de dados.<br>Falha na conexão com o servidor que despacha os e-mails.  |
| <b>Saída</b>         |   |
| <b>Pré-condições</b> | É obrigatória a existência de um usuário responsável para concretizar o processo.   |
| <b>Pós-condições</b> | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – enviou o e-mail com sucesso.<br>1 – falha no envio do e-mail.  |

F.11 Tabela das características do contrato de classe desenvolvidos para a operação **atualizar dados no servidor aplicação()** da classe Servidor de aplicação.

Tabela F.11 – Contrato da classe Servidor de aplicação, operação: atualizar dados no servidor aplicação().

|                          |  |
|--------------------------|--|
| <b>Classe</b>            | Servidor de aplicação.   |
| <b>Nome operação</b>     | atualizar dados no servidor aplicação()  |
| <b>Responsabilidades</b> | O método em questão aciona funcionalidades do servidor de aplicação para realizar a tarefa de atualizar os dados que são tratados pelo gerenciador de serviços. Uma API estabelece o relacionamento entre o gerenciador de serviços com o servidor de aplicação (figura 3.2).  |
| <b>Tipo</b>              | Operação de classe.  |
| <b>Referência</b>        | Caso de uso: (Apêndice A-a) cadastramento de serviços; (Apêndice A-c) cadastramento da interface do serviço; (Apêndice A-g) alteração do estado do serviço.  |
| <b>Observações</b>       |  |
| <b>Exceções</b>          | Falha na conexão com o servidor de aplicação.  |
| <b>Saída</b>             |  |
| <b>Pré-condições</b>     | Ter a instalação e a configuração da comunicação entre o gerenciador de serviços e o servidor de aplicação.  |
| <b>Pós-condições</b>     | O atributo código de retorno é atualizado com os seguintes valores:<br>0 – executou a submissão de atualização com sucesso.<br>1 – falha na execução da submissão da atualização.<br>10 – executou a submissão de atualização da interface do serviço com sucesso.<br>11 – falha na execução da submissão da atualização da interface do serviço.<br>20 – executou a alteração do estado do serviço com sucesso.<br>21 – falha na execução da submissão da atualização do estado do serviço. |

## Apêndice G –Detalhes da especificação dos casos de teste.

As especificações detalhadas dos contratos das classes (Apêndice F) são fundamentais para a elaboração das especificações dos casos de teste. Estas especificações dos casos de teste são organizadas e apresentadas na forma dos itens G.1 até G.11 que apresentam, em detalhes o objetivo de cada teste, quais são as condições necessárias para viabilizá-lo e o procedimento a ser executado para realizá-lo.

### G.1 Teste do retorno após a chamada da função: **registrar dados do serviço()**.

#### a) Objetivo do teste

Validar as condições do registro de um serviço após a execução da operação: registrar dados do serviço().

#### b) Condições

| # | Condições  |
|---|--|
| 1 | O serviço “autenticação de usuário” não está cadastrado. |

#### c) Procedimento de teste

| # | Procedimento de teste  | Resultado esperado                             |
|---|--|--|
| 1 | Informar os seguintes conteúdos para os campos que compõem os argumentos da operação, versão: “1.0.2”, nome: “Autenticação de usuário”, descrição: “Este serviço oferece uma validação de autenticação de usuário.”, responsável: “Rodrigues Alves”, projeto: “Funcionalidades globais.”, plataforma operacional: “windows”, tipo do serviço: “”, data inclusão no registro: “20/04/2005 11:03”, data última modificação: “”, data desativação: “”, data expiração: “”, responsável pela homologação: “Rodrigues Alves”, data homologação: “01/04/2005”, fornecedor: “”, contrato: “”, data de vencimento do contrato: “”, linguagem de programação: “VB6”, configuração: “Através de uma pasta que contém o programa SETUP.EXE é possível instalar o serviço”, criticidade: “Alta”, local instalação: “d:\aplicativos\pa\pa0200\”, data instalação: “20/04/2005 11:03”, período de atividade: “8X5”, tempo fora do ar: “”, serviço substituído: “”, versão serviço substituído: “”, serviço utilizado: “”, versão serviço utilizado: “”, estado: “Ativo”.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. |

### G.2 Teste do retorno após a chamada da função: **verificar serviços vencidos()**.

#### a) Objetivo do teste

Validar as condições após a execução da operação: verificar serviços vencidos().

## b) Condições

| # | Condições   |
|---|---|
| 1 | Existem serviços cadastrados que estão vencidos, estes estão atribuídos a um período de tempo que compreende desde uma data inicial, por exemplo, “01/01/1900” até uma data final que é anterior a um dia da data corrente. |

## c) Procedimento de teste

| # | Procedimento de teste  | Resultado esperado   |
|---|--|--|
| 1 | Informar os seguintes valores para os argumentos da operação: “01/01/1900” e a data corrente do dia.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. Consequentemente ao resultado zero, é gerada uma lista para consulta que contém os serviços vencidos. |

G.3 Teste do retorno após a chamada da função: **verificar serviços a vencer()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: verificar serviços a vencer().

## b) Condições

| # | Condições   |
|---|---|
| 1 | Existem serviços cadastrados que futuramente vencerão, estes estão atribuídos a um período de tempo que compreende desde a data corrente do dia até uma data final que pode ser de 3 meses no futuro. |

## c) Procedimento de teste

| # | Procedimento de teste  | Resultado esperado   |
|---|--|--|
| 1 | Informar os seguintes valores para os argumentos da operação: data corrente do dia e data corrente do dia acrescida de três meses.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. Consequentemente ao resultado zero, é gerada uma lista para consulta que contém os serviços a vencer dos próximos três meses. |

G.4 Teste do retorno após a chamada da função: **registrar dados da interface()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: registrar dados da interface().

## b) Condições

| # | Condições  |
|---|--|
| 1 | O serviço “autenticação de usuário” está persistido pela classe Serviço. |
| 2 | Não foi cadastrada a interface do serviço.                               |

## c) Procedimento de teste

| # | Procedimento de teste  | Resultado esperado                             |
|---|--|--|
| 1 | Informar os seguintes conteúdos para os campos que compõem os argumentos da operação, serviço: “autenticação de usuário”, versão: “1.0.2”, nome 1º. argumento: “usuário”, descrição: “Nome do usuário para o login”, tipo: “caractere”, tamanho: “20”, direção: “in”, observação: “”, nome 2º. argumento: “senha”, descrição: “Senha associada ao usuário de login”, tipo: “caractere”, tamanho: “8”, direção: “in”, observação: “”.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. |

G.5 Teste do retorno após a chamada da função: **registrar dados do serviço()**.

## a) Objetivo do teste

Validar as condições do registro do componente utilizado por um serviço após a execução da operação: registrar dados do serviço().

## b) Condições

| # | Condições  |
|---|--|
| 1 | O serviço “autenticação de usuário” está persistido pela classe Serviço. |
| 2 | Não foram cadastrados os componentes utilizados pelo serviço.            |

## c) Procedimento de teste

| # | Procedimento de teste   | Resultado esperado                             |
|---|---|--|
| 1 | Informar os seguintes conteúdos para os campos que compõem os argumentos da operação, versão: “3.70.08.20”, nome: “ <i>Driver ODBC para Sql Server</i> ”, descrição: “Este <i>driver</i> permite o acesso aos bancos de dados Microsoft SQL Server.”, responsável: “Microsoft”, projeto: “ <i>Drivers</i> ”, plataforma operacional: “windows”, tipo do serviço: “ <i>Drivers</i> ”, data inclusão no registro: “20/04/2000 11:03”, data última modificação: “”, data desativação: “”, data expiração: “”, responsável pela homologação: “Microsoft”, data homologação: “”, fornecedor: “Microsoft”, contrato: “”, data de vencimento do contrato: “”, linguagem de programação: “C”, configuração: “”, criticidade: “Alta”, local instalação: “c:\windows\system32\sqlsrv32.dll, data instalação: “20/04/2000 11:03”, período de atividade: “24X7”, tempo fora do ar: “”, serviço substituído: “”, versão serviço substituído: “”, serviço utilizado: “autenticação de usuário”, versão serviço utilizado: “1.0.2”, estado: “Ativo”.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. |

G.6 Teste do retorno após a chamada da função: **registrar modificação()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: registrar modificação().

## b) Condições

| # | Condições  |
|---|--|
| 1 | O serviço “autenticação de usuário” está no processo de ser persistido pela classe Serviço.  |
| 2 | Não foi executado o registro de modificação que se refere a inclusão do serviço “autenticação de usuário”, devido a condição anterior. |

## c) Procedimento de teste

| # | Procedimento de teste   | Resultado esperado                             |
|---|---|--|
| 1 | Informar os seguintes conteúdos para os campos que compõem os argumentos da operação, serviço: “autenticação de usuário”, versão: “1.0.2”, data modificação: “20/04/2005 11:03”, tipo de modificação: “Inc-Serviço”, descrição modificação: “- informar o atributo e o conteúdo de todos os campos que participam da inclusão -”, responsável: “Rodrigues Alves”.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. |

G.7 Teste do retorno após a chamada da função: **registrar dados do problema()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: registrar dados do problema().

## b) Condições

| # | Condições  |
|---|--|
| 1 | Não ter o problema de conflito de componentes de um determinado serviço já cadastrado. |

## c) Procedimento de teste

| # | Procedimento de teste   | Resultado esperado                             |
|---|---|--|
| 1 | Informar os seguintes conteúdos para os campos que compõem os argumentos da operação, serviço: “autenticação de usuário”, versão: “1.0.2”, data problema: “30/04/2005 15:00”, situação: “em aberto”, data da constatação “30/04/2005 11:15”, criticidade: “Baixa”, tipo: “conflito com os componentes do ambiente”, responsável “Otreblada Yren”, descrição: “Durante a instalação do serviço o componente de login entrou em conflito com o componente já registrado no ambiente”.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. |

G.8 Teste do retorno após a chamada da função: **verificar problemas não concluídos()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: verificar problemas não concluídos().

## b) Condições

| # | Condições  |
|---|--|
| 1 | Existem problemas cadastrados que não estão concluídos, ou seja, estão pendentes. O período de tempo destes problemas compreende uma faixa que se inicia, por exemplo, em “01/01/1900” e tem como data final a data corrente do dia. |

## c) Procedimento de teste

| # | Procedimento de teste   | Resultado esperado  |
|---|---|---|
| 1 | Informar os seguintes valores para os argumentos da operação:<br>“01/01/1900” e a data corrente do dia.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. Conseqüentemente ao resultado zero, é gerada uma lista para consulta que contém os problemas não concluídos. |

G.9 Teste do retorno após a chamada da função: **registrar dados do responsável()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: registrar dados do responsável().

## b) Condições

| # | Condições  |
|---|--|
| 1 | Não ter o usuário (administrador ou desenvolvedor) responsável cadastrado. |

## c) Procedimento de teste

| # | Procedimento de teste  | Resultado esperado                             |
|---|--|--|
| 1 | Informar os seguintes conteúdos para os campos que compõem os argumentos da operação, nome: “Rodrigues Alves”, e-mail: “ <a href="mailto:ralves@brazil.com.br">ralves@brazil.com.br</a> ”, telefone: “11 444-5555”, tipo de responsabilidade: “Desenvolvedor” e se o usuário deseja receber e-mails.<br>Submeter a execução da operação. | O código de retorno da operação executada é 0. |

G.10 Teste do retorno após a chamada da função: **avisar responsável()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: avisar responsável().

## b) Condições

| # | Condições  |
|---|--|
| 1 | Ter um serviço, por exemplo, “autenticação de usuário” com o estado atual ativo. |

## c) Procedimento de teste

| # | Procedimento de teste  | Resultado esperado   |
|---|--|--|
| 1 | Através do programa “Altera Estado do Serviço” editar o serviço e mudar o valor da propriedade estado para inativo e, subseqüentemente, pressionar o botão salvar. | O código de retorno da operação executada é 0. Portanto, um e-mail contendo o descritivo desta mudança é enviado a todos os desenvolvedores de serviços para torná-los conscientes da intervenção. |

G.11 Teste do retorno após a chamada da função: **atualizar dados no servidor aplicação()**.

## a) Objetivo do teste

Validar as condições após a execução da operação: atualizar dados no servidor aplicação().

## b) Condições

| # | Condições  |
|---|--|
| 1 | Ter um serviço, por exemplo, “autenticação de usuário” com o estado atual ativo. |

## c) Procedimento de teste

| # | Procedimento de teste  | Resultado esperado  |
|---|--|---|
| 1 | Através do programa “Altera Estado do Serviço” editar o serviço e mudar o valor da propriedade estado para inativo e, subseqüentemente, pressionar o botão salvar. | O código de retorno da operação executada é 20. Portanto, através de uma chamada de uma API o serviço em questão é desativado no servidor de aplicação. |

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)



[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)