

**RODRIGO GIOIA GUIMARÃES
TAKAE TERASE NAGAMINE
VITOR HUGO SUZUKI**

**SISTEMA DE INFORMAÇÃO PARA COMPARTILHAMENTO DE
BASES DE DADOS DISTRIBUÍDAS DE BIODIVERSIDADE**

**Projeto de Formatura apresentado à disciplina
PCS 2050 – Projeto de Formatura II, da
Escola Politécnica da Universidade de São
Paulo.**

**São Paulo
2005**

**RODRIGO GIOIA GUIMARÃES
TAKAE TERASE NAGAMINE
VITOR HUGO SUZUKI**

**SISTEMA DE INFORMAÇÃO PARA COMPARTILHAMENTO DE
BASES DE DADOS DISTRIBUÍDAS DE BIODIVERSIDADE**

**Projeto de Formatura apresentado à disciplina
PCS 2050 – Projeto de Formatura II, da
Escola Politécnica da Universidade de São
Paulo.
Área do Projeto: Arquiteturas de Sistemas
Distribuídos.
Orientador: Prof. Dr. Pedro Luiz Pizzigatti
Corrêa**

**São Paulo
2005**

**AUTORIZAMOS A DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO
DESDE, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO PARA
FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.**

FOLHA DE APROVAÇÃO

Rodrigo Gioia Guimarães

Takae Terasse Nagamine

Vitor Hugo Suzuki

Sistema de Informação para Compartilhamento de Bases de Dados Distribuídas de
Biodiversidade

Projeto de Formatura apresentado à disciplina PCS 2050 –
Projeto de Formatura II, da Escola Politécnica da
Universidade de São Paulo.

Área do Projeto: Arquiteturas de Sistemas Distribuídos

Aprovado em:

Banca Examinadora

Prof Dr. _____

Instituição _____ Assinatura _____

Prof Dr. _____

Instituição _____ Assinatura _____

Prof Dr. _____

Instituição _____ Assinatura _____

Prof Dr. _____

Instituição _____ Assinatura _____

Prof Dr. _____

Instituição _____ Assinatura _____

AGRADECIMENTOS

Ao Prof. Doutor Pedro Luiz Pizzigatti Corrêa da Escola Politécnica da Universidade de São Paulo, pela atenção e apoio durante o processo de definição e orientação.

Ao Prof. Doutor Antônio Mauro Saraiva da Escola Politécnica da Universidade de São Paulo, pela atenção e apoio em todo o processo de desenvolvimento do projeto.

Ao Mestrando Marcelo Succi de Jesus Ferreira da Escola Politécnica da Universidade de São Paulo, pelo apoio no processo de definição da Arquitetura de Segurança.

À Doutoranda do Programa de Pós Graduação da Escola Superior de Agricultura Luiz de Queiroz - ESALQ/USP Mariana Carvalhaes, pelo apoio no processo de estudo e mapeamento dos dados das diversas espécies de Bromélias.

Ao Pesquisador Carlos Paniago da Embrapa Monitoramento por Satélite, pelo apoio e auxílio no processo de estudo e desenvolvimento do Provedor de Mapas de Satélites.

Ao Departamento de Engenharia da Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo, por colocar a disposição a área experimental e o laboratório.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo financiamento das bolsas de iniciação científica 04/01110-5 e 04/01111-1.

À Embrapa Monitoramento por Satélite por colocar a disposição os mapas de satélites e sua área experimental.

RESUMO

Este estudo trata do desenvolvimento de um Sistema de Informação para Compartilhamento de Bases de Dados Distribuídas de Biodiversidade de acordo com padronizações internacionais. Através da utilização destes para catalogação de espécies, pode-se criar uma articulação entre os diferentes segmentos envolvidos, para estimular a troca de informação técnico-científica e assim ajudar no combate ao extrativismo ilegal e a biopirataria. Também será levantado um conjunto de requisitos e estratégias de segurança para o projeto de Sistemas Distribuídos que permitam integrar bases de dados de biodiversidade e que gerenciam o acesso a informações sobre nome, classificação e distribuição de espécies de plantas, animais e microrganismos, cujo acesso completo é fundamental para a definição de políticas públicas responsáveis pela preservação da biodiversidade de um país.

Para este trabalho propõe-se uma Arquitetura que utiliza computação Orientada a Serviços e o uso de um padrão internacional, o ABCD (ABCD-a, 2005). Para implementar a arquitetura proposta foram consideradas as seguintes tecnologias: Linguagem de programação Java (JAVA, 2005) e os *containers* JSP e *Servlets*, usando *Tomcat* (TOMCAT, 2004); Ambiente de Desenvolvimento *open source* para teste de programas Java; o Ambiente de Desenvolvimento Integrado (IDE) Eclipse (Eclipse, 2005) e as diretrizes de *Extreme Programming* (XP, 2005) para o desenvolvimento dos softwares

Todo esse estudo visa fornecer um amplo conhecimento de arquiteturas de segurança de sistemas distribuídos, baseadas em *Web Services* (STAL, 2002) e em padrões internacionais de catalogação ABCD (ABCD-b, 2005) e DarwinCore (DarwinCore, 2004) permitindo assim o desenvolvimento de um sistema, que enfoca o compartilhamento seguro de informações entre pesquisadores relacionados ao estudo de bromélias.

PALAVRAS-CHAVES: Padrão ABCD, Informática e Biodiversidade; Banco de Dados Distribuídos; *Web Services* (W3C-a, 2005).

ABSTRACT

This study is about the development of a Service Oriented Architecture to share Biodiversity Information System using Distributed Databases, based on international standards. This system can create a relationship among the different organizations involved, allowing the exchange of technical and scientific information that can help to combat the bio-piracy and the illegal extractivism. In this study will be considered the security requirements to Distributed Systems that allow the integration of distributed databases, which manage information about name, classification, distribution of species of plants, animals and microorganisms. The complete access is necessary to define politics that will be responsible for the preservation of the biodiversity of the country.

In this study is proposed an Architecture based on Services and the use of an international standard, the ABCD (ABCD-a, 2005). In order to develop the architecture it was proposed the following technologies: Java language (JAVA, 2005) and the containers JSP and Servlets, using Tomcat (TOMCAT, 2004), Open Source Development Environment to do the program tests, Integrated Development Environment (IDE) Eclipse (Eclipse, 2005) and some best practices of the Extreme Programming (XP, 2005) to the development of the software.

This study will supply a great knowledge of security architectures of distributed systems, based on Web Services (Stal, 2005) and the international standards to catalogue species (ABCD (ABCD-a, 2005) and DarwinCore (DARWINCORE, 2005)) allowing the development of a system that has as main target the exchange of information among researchers of Bromelias.

Key Words: ABCD Standard, Informatics and Biodiversity, Distributed Databases, Web Services (W3C-a, 2005).

LISTA DE ILUSTRAÇÕES

Figura 1: Camadas de Serviços.....	22
Figura 2: Arquitetura utilizada por <i>Web Services</i>	25
Figura 3: Arquitetura do DiGIR.....	27
Figura 4: Arquitetura Básica.....	35
Figura 5: Arquitetura Segurança.....	37
Figura 6: Comunicação entre entidades	38
Figura 7: Autenticação entre entidades	39
Figura 8: Serviços compostos.....	40
Figura 9: Definição de Serviços no Sistema.....	41
Figura 10: Arquitetura Baseada em Componentes	45
Figura 11: Implementação da arquitetura.....	46
Figura 12: Diagrama de Classes do Portal de Bromélias	61
Figura 13: Diagrama de Classes do Provedor de Bromélias	62
Figura 14: Diagrama de Classes do Provedor de Mapas.....	63
Figura 15: Diagrama de Componentes do Sistema de Bromélias	64
Figura 16: Diagramas de Seqüência do Provedor de Bromélias - Classe <i>Coordinator</i> recebendo autenticação inválida e válida	65
Figura 17: Diagramas de Seqüência do Provedor de Bromélias - Classe <i>Coordinator</i> recebendo autorização válida e inválida.....	66
Figura 18: Diagramas de Seqüência do Provedor de Bromélias - Classe <i>Coordinator</i> acessando serviço do provedor de mapas.....	67
Figura 19: Diagramas de Seqüência do Provedor de Bromélias - Classe <i>Coordinator</i> acessando serviço do provedor de mapa. Erro ao acessar serviço.....	68
Figura 20: Diagramas de Seqüência do Provedor de Bromélias - Classe <i>Coordinator</i> executando transação do provedor de Bromélias.....	69
Figura 21: Diagramas de Seqüência do Provedor de Bromélias - <i>ShowMapInfo</i> acessando serviço do provedor de mapa.	70
Figura 22: Diagramas de Seqüência do Portal de Bromélias acessando serviço do provedor de Bromélias.....	71
Figura 23: Diagramas de Seqüência do Provedor de Mapas - Classe <i>GetMap</i> executando o serviço de envio de mapas.	72
Figura 24: Diagrama Entidade Relacionamento da Base de Dados	73
Figura 25: Interface de <i>Login</i>	74
Figura 26: Interface de Consulta.....	75
Figura 27: Interface de Consulta de Imagem de Satélite	76

LISTA DE TABELAS

Tabela 1: Mapeamento das funcionalidades da Arquitetura SOA.....	41
Tabela 2: Tabela Comparativa das Topologias	47

SUMÁRIO

1. INTRODUÇÃO	12
1.1 VISÃO GERAL	12
1.2 MOTIVAÇÃO	14
1.3 JUSTIFICATIVA	14
1.4 OBJETIVOS GERAIS	16
1.5 OBJETIVOS ESPECÍFICOS	16
1.6 ESTRUTURA DO TRABALHO	17
2. REVISÃO DA LITERATURA	18
2.1 MÉTODOS DE DESENVOLVIMENTO	20
2.1.1 <i>Especificação</i>	20
2.1.1.1 UML	20
2.1.2 <i>Projeto</i>	21
2.1.2.1 XP	21
2.1.2.2 <i>Computação Orientada a Serviços</i>	21
2.1.2.3 <i>Web Services</i>	23
2.1.2.4 <i>DiGIR</i>	25
2.2 FERRAMENTAS DE DESENVOLVIMENTO	28
2.2.1 <i>Linguagens de programação</i>	28
2.2.1.1 J2EE	28
2.2.1.2 JSP (Java Server Page)	28
2.2.2 <i>Ambiente de Desenvolvimento</i>	28
2.2.2.1 Eclipse	28
2.2.2.2 Axis	29
2.2.2.3 CVS	29
2.2.2.4 Tomcat	29
2.2.2.5 PostgreSQL	29
2.2.2.6 MapServer	30
2.2.3 <i>Protocolo de Segurança</i>	31
2.2.3.1 LDAP	31
2.2.3.2 Open LDAP	32
2.2.3.3 JNDI	32
2.3 PADRÕES INTERNACIONAIS	33
2.3.1 ABCD	33
2.3.2 <i>Darwin Core</i>	33
2.4 CONCLUSÃO	34
3. ARQUITETURA DO SISTEMA	35
3.1 SISTEMA DE INFORMAÇÃO	35
3.2 SISTEMA DE SEGURANÇA	37
3.2.1 <i>Estudo de Caso: Usuário A acessando serviços a partir do Portal de Bromélia</i>	37
3.2.2 <i>Estudo de Caso: Comunicação entre entidades dentro de uma comunidade</i>	38
3.2.3 <i>Estudo de Caso: Autenticação entre entidades dentro de uma comunidade</i>	39
3.2.4 <i>Estudo de Caso: Serviços Compostos</i>	39
3.3 SISTEMA DE INFORMAÇÃO - ARQUITETURA SOA	40
3.3.1 <i>Projeto do Sistema</i>	42
3.3.1.1 Base de Dados	42
3.3.1.2 Provedor	43
3.3.1.3 Portal	44

3.4	SISTEMA DE INFORMAÇÃO - ARQUITETURA BASEADA EM COMPONENTES.....	44
3.5	SISTEMA DE INFORMAÇÃO - TECNOLOGIAS UTILIZADAS.....	45
4	IMPLEMENTAÇÃO	49
5.	DISCUSSÃO DOS RESULTADOS	53
5.1	RESULTADOS OBTIDOS	53
5.2	SUGESTÃO DE CONTINUIDADE DO PROJETO.....	53
6.	BIBLIOGRAFIA	56
	ANEXO A: DIAGRAMAS DE CLASSES.....	61
	ANEXO B: DIAGRAMA DE COMPONENTES	64
	ANEXO C: DIAGRAMA DE SEQUÊNCIA - PROVEDOR DE BROMÉLIAS	65
	ANEXO D: DIAGRAMA ENTIDADE RELACIONAMENTO.....	73
	ANEXO E: INTERFACE DO SISTEMA.....	74
	APÊNDICE A: DESCRIÇÃO DO CD DO PROJETO DE FORMATURA	77

1. INTRODUÇÃO

1.1 Visão Geral

A Mata Atlântica apresenta uma grande diversidade ambiental decorrente de sua ampla variação latitudinal e altitudinal, com cerca de trinta graus no limite norte-sul e áreas desde o nível do mar até quase três mil metros de altitude no topo das serras. Tal amplitude de domínios determina a sua principal característica que é a diversidade de espécies, assim esta se constitui no segundo maior bloco de floresta tropical do país com 20.000 espécies de plantas, das quais 8.000 são exclusivas (SOS Mata Atlântica, 2005).

Cerca de 70% da população vive no domínio atlântico e este processo de ocupação deu-se desde o início de forma predatória do ponto de vista ecológico, social e econômico. Este fato infelizmente coloca a Mata Atlântica entre os cinco primeiros na lista dos *Hot Spots*, as áreas de alta biodiversidade mais ameaçadas do planeta e prioritárias do planeta para ações urgentes de conservação, com apenas 7,3% de área remanescente.

A costa leste brasileira é considerada centro de diversidade de várias famílias botânicas, dentre estas a das bromélias. Com cerca de 25000 espécies e 42 gêneros, *Bromeliaceae* é uma das maiores entre as angiospermas. Exceto por uma espécie de *Pitcairnia*, que ocorre no oeste africano, é restrita às Américas, distribuindo-se desde a Argentina e Chile até o norte do estado da Virgínia, nos EUA. Estima-se que 40% de espécies e 73% de gêneros ocorram no Brasil, sendo que muitos gêneros e espécie são endêmicos, i.e., ocorrem exclusivamente nas formações vegetais do domínio atlântico como, por exemplo, *Canistrum*, *Nidularium* e *Wittrockia*. A elevada taxa de endemismo pode ser traduzida da seguinte forma: duas em cada três espécies de bromélias ocorrem exclusivamente na Floresta Atlântica, o que aumenta consideravelmente o risco de extinção, seja pelo desmatamento, seja pelo extrativismo predatório (Coffani Nunes, 2002).

Não há dados oficiais sobre o extrativismo de bromélias na Mata Atlântica, mas sabe-se que sua prática se dá em diferentes regiões ameaçando espécies como é o caso de *Alcantarea imperialis* na região da Serra dos Órgãos. De maneira geral, as rodovias que a cortam são importantes vias de acesso para extrativistas sendo muitas vezes também pontos de comercialização. A região entre o Rio de Janeiro e São Paulo com remanescentes em bom

estado de preservação e grande diversidade de espécies é estratégica pela proximidade de grandes centros consumidores. Em remanescentes florestais do sul de São Paulo e norte do Paraná a pressão está sendo impulsionada pela duplicação da BR-116 (Coffani Nunes, 2002).

A reversão deste panorama depende de vários fatores como a existência de políticas de conservação e uso de seus recursos, a eficiência do sistema de fiscalização e da geração, difusão e aplicação do conhecimento técnico científico existente.

Já do outro lado da cadeia, a demanda de bromélias para paisagismo é crescente por estas apresentarem características que vêm a resolver problemas do paisagismo moderno como uma arquitetura peculiar, a facilidade de adaptação, a rusticidade que permite pouca manutenção e pouco espaço de terra. Em decorrência disso, hoje há horticultores investindo na sua produção e comercialização, mas o número de produtores não é suficiente para suprir a demanda, o que provoca um considerável aumento no extrativismo ilegal.

A falta de articulação entre as diferentes áreas envolvidas, associada às lacunas no conhecimento biológico e de aspectos socioeconômicos do extrativismo predatório ou não, culmina na ausência de princípios e critérios confiáveis que subsidiem a ação eficiente dos órgãos oficiais na fiscalização ou no licenciamento de atividades econômicas que tenham como princípio a compatibilidade entre a atividade econômica e a conservação da floresta.

O interesse é crescente na utilização na comunidade científica e leiga, principalmente por parte de colecionadores, pesquisadores, comerciantes e produtores, que representam os diferentes segmentos envolvidos na comercialização de bromélias provenientes do cultivo e extração. Há a necessidade de criar uma articulação entre estes elos através de ações que estimulem a troca de informações técnico-científicas e a elaboração de uma análise mais ampla e real da questão. Espera-se que o desenvolvimento deste sistema de informações disponibilize um espaço para compartilhamento destas informações e da discussão dos diversos aspectos envolvidos na conservação e uso das bromélias, um produto florestal não madeireiro de rara beleza e que é, ao mesmo tempo, um agente e um indicador da diversidade de espécies e de nichos ecológicos que compõem a Mata Atlântica brasileira.

1.2 Motivação

A busca pela compreensão, proteção e uso controlado da biodiversidade terrestre pode ser auxiliada através de um rígido controle da localização de áreas que contêm as espécies em estudo. Sob a pressão de um crescimento populacional intenso e de uma rápida expansão e globalização das economias, toda a esfera da biodiversidade (o que inclui a singela porção associada às bromélias) tem sido amplamente ameaçada a taxas alarmantes. Entre as espécies sujeitas as atividades exploratórias exacerbadas, poderíamos encontrar a chave para muitos dos enigmas e problemas que permeiam o mundo atual (novos princípios ativos, medicamentos alternativos, proteção do próprio meio-ambiente).

O crescente interesse da comunidade científica e leiga, principalmente por parte dos colecionadores, pesquisadores, comerciantes e produtores, faz se necessário à criação de uma base de dados que disponibilize as informações relativas à conservação e ao uso das bromélias. Desta forma, há o compartilhamento das informações técnico-científicas, possibilitando a obtenção de princípios e critérios confiáveis que subsidiem a ação eficiente dos órgãos oficiais na fiscalização ou no licenciamento de atividades econômicas que tenham como princípio à compatibilidade entre a atividade econômica e a conservação da floresta.

Para a proteção destes meios, seria interessante apontar especificamente onde as espécies poderiam ser encontradas (mapeamento da localização geográfica), sendo, dessa forma, mais facilmente conservadas, uma vez que, na maioria das vezes, a negligência e falta de conhecimento das regiões devastadas em potencial, aumenta cada vez mais o índice de espécies desaparecidas.

1.3 Justificativa

Um Sistema de Informação para Compartilhamento de Bases de Dados Distribuídas de Biodiversidade que atenda as necessidades consideradas anteriormente deve catalogar as espécies de bromélias, disponibilizando as informações de acordo com o perfil de utilização de cada agente envolvido. Para isso é fundamental o uso de Tecnologia de Informação, especialmente voltada para a WEB, que permita também a integração com outras bases de dados de biodiversidade.

Atualmente várias iniciativas isoladas vêm sendo desenvolvidas no sentido de catalogar bases de dados de espécies brasileiras. Dentre os projetos com esse objetivo, pode-se destacar alguns projetos nacionais, tais como o *SimBiota* (FAPESP, 2005), Sistema de Informações responsável por catalogar dados sobre espécies e recursos naturais do Estado de São Paulo, identificados por pesquisadores que participam do Projeto Biota da FAPESP (Fundação de Amparo à pesquisa do Estado de São Paulo); o *WebBee* (WEBBEE, 2005), sistema responsável por catalogar informações sobre biodiversidade brasileira de abelhas nativas.

No exterior podem-se destacar vários projetos em andamento: na América do Norte, MaNIS (*Mammal Network Information System*) (Berkeley, 2005), na Europa (*European Natural History Specimen Information Network*) e na Austrália (AVH – *Australia's Virtual Herbarium*) (AVH, 2005). Porém, cada um desses sistemas atende às necessidades da área de domínio de informação de biodiversidade para o qual foram desenvolvidos, não suportando a inclusão de informações específicas geradas em atividades de pesquisa da bromélia, além de não considerarem aspectos envolvendo a interação entre diferentes agentes considerados: pesquisadores, formuladores de políticas de conservação, produtores e colecionadores.

A integração de informações entre bases de dados de biodiversidades existentes envolve a aplicação de padrões internacionais. Dentre as entidades que vem trabalhando com essa questão, destaca-se o GBIF (*Global Biodiversity Information Facility*). O GBIF é uma iniciativa internacional responsável pela coordenação de uma rede que pretende integrar e disponibilizar na Internet informações de coleções biológicas do mundo todo (cerca de três bilhões de exemplares) e uma lista eletrônica com o nome e a classificação taxonômica de 1,75 milhões de espécies descritas cientificamente, o chamado catálogo da vida. GBIF vem estabelecendo padronizações que definem o conteúdo das informações para as bases de dados de biodiversidade (ABCD-a, 2005) além de padronização para acesso de informações entre sistemas de biodiversidade, baseado em *Web Services* denominado *DiGIR* (DIGIR, 2005).

O tratamento e o estudo de imagens do globo terrestre via satélites já vem sendo feito por entidades como a Embrapa (Empresa Brasileira de Pesquisa Agropecuária) (EMBRAPA, 2005), por sua área denominada Centro Nacional de Pesquisa de Monitoramento por Satélite. Atualmente a Embrapa atua viabilizando soluções tecnológicas, competitivas e sustentáveis para o agro negócio, baseadas na utilização do monitoramento por satélite na agropecuária, com aplicações do sensoriamento remoto, geoprocessamento e tecnologias de informação, em benefício da sociedade. Contudo, o estudo da ocorrência de bromélias na Mata Atlântica e a

biodiversidade que ocorre ao seu redor baseados em mapas de satélites, ainda é inexistente, criando uma oportunidade para que seja desenvolvido um sistema que não apenas faça o estudo da biodiversidade ao redor, como também cruze os dados com outras regiões, permitindo obter a possível localização de bromélias na mata atlântica Brasileira. A criação de mapas de regiões específicas mostrando a distribuição mais provável das espécies de bromélias na natureza seria extremamente valiosa para atividades como planejamento de expedições controladas e viabilização de programas locais para aprimorar a conscientização e conservação das regiões. Tais projeções poderiam ser eficazmente utilizadas para guiar a coleta de espécies para fins acadêmicos e científicos, para aprofundar o estudo taxonômico e genético de certos exemplares, além de focar o mapeamento da distribuição de outras espécies e organismos que permeiam o meio comum às bromélias. A precisão e nível de detalhe dos mapas poderiam eliminar a lentidão e o alto custo envolvido no processo de mapeamento manual das espécies encontradas atualmente. Logo, esta arquitetura tem como objetivo o desenvolvimento de um produto, que não apenas atenda os pesquisadores voltados ao estudo de bromélias, como também os que queiram cadastrar e catalogar outras espécies de Biodiversidade. Isto é garantido pela grande abrangência e flexibilidade da base de dados, além da arquitetura baseada em Componentes e Serviços, que permite a interoperabilidade de sistemas com tecnologias diferentes.

1.4 Objetivos Gerais

- Desenvolver mecanismos de Segurança que permitam compartilhamento de Bases de Dados Distribuídas de Biodiversidade.
- Desenvolver um sistema utilizando a WEB para gerenciamento de informações das bromélias, de acordo com padronizações internacionais para catalogação e integração de informações de biodiversidade.

1.5 Objetivos Específicos

- Gerar ambiente de desenvolvimento de software baseado na Arquitetura de Sistemas J2EE (*Java Second Enterprise Edition*) (J2EE, 2005) e que incorpore algumas diretrizes práticas de XP (*Extreme Programming*) (XP, 2005).

- Estudar arquiteturas de sistemas distribuídos de segurança que permitam a autenticação e a autorização para acessar os serviços e as informações disponíveis, baseado na arquitetura de *Web Services* (STAL, 2005);
- Especificação e implementação de uma Arquitetura de um Sistema de Informação baseada em serviços e que implemente uma base de dados utilizando padrão internacional para informação de espécies, denominada ABCD (ABCD-a, 2005);
- Identificar e implementar os principais serviços do sistema de forma atender as necessidades de manipulação de dados dos usuários com perfil de pesquisador;
- Estudar, especificar e implementar a transferência e tratamento de imagens e mapas via *Web-Services*.

1.6 Estrutura do trabalho

Para melhor compreensão, este documento será dividido em:

- **Revisão da Literatura:** Conceitos, metodologias e ferramentas utilizados para o estudo, modelagem e elaboração do sistema final. São explicitadas suas principais características e contribuições para o projeto;
- **Arquitetura do sistema:** Específico sobre o processo de modelagem do sistema. Considera as principais técnicas utilizadas, características particulares, vantagens e motivações;
- **Implementação:** Referente às etapas de implementação do sistema desenvolvido segundo a modelagem previamente realizada.
- **Discussão dos Resultados:** Análise sobre o desenvolvimento do projeto considerando aspectos de impacto e viabilidade. Reflete-se também a respeito das prováveis contribuições deste projeto;
- **Anexos:** Detalhamento adicional de informações específicas (diagramas, interfaces e DER).

2. REVISÃO DA LITERATURA

A seguir, estão descritos os conceitos, as técnicas e ferramentas que foram aplicadas na pesquisa:

- Técnica de Modelagem de Software baseada em UML (*Unified Modelling Language*) (RUMBAUGH et al., 2003), para a modelagem dos programas em java do sistema. O UML é composto por vários diagramas que representam visões diferentes do sistema.
- Técnicas de Engenharia de Software, baseadas em diretrizes de *Extreme Programming* (XP) para estabelecer método de trabalho durante o desenvolvimento (XP, 2005). O XP é baseado em valores de simplicidade, comunicação e feedback.
- Computação orientada a Serviços que estabelece os conceitos básicos da Arquitetura proposta.
- *Web Services* para integração de Sistemas. Essa Arquitetura é baseada em conceitos de *Middlewares* Orientados a Objetos e Computação Orientada a Serviços, que foram aplicados a Internet. Através desta arquitetura é possível o desenvolvimento de sistemas distribuídos agregando serviços e informações disponíveis em vários sistemas, permitindo a construção de portais e o desenvolvimento de aplicações que envolvam a integração de negócios e troca de informações entre diferentes bases de dados.
- DiGIR para compartilhamento de informações com outros sistemas. Esta arquitetura estabelece os princípios para integração de bases de dados distribuídas de Biodiversidade, permitindo consultar e rastrear informações em bancos de dados distintos e apresentar os resultados ao usuário como se as informações tivessem origem numa base de dados única;
- Linguagem de programação Java (JAVA, 2005) e os *containers* JSP (*Java Server Pages*) (JSP, 2004) usando Tomcat (TOMCAT, 2005). Trata-se de uma tecnologia para a criação de aplicações na Internet. Ela é responsável pelo controle dos dados trocados entre o computador do cliente e o servidor e vice-versa.

- Eclipse (ECLIPSE, 2004), Ambiente de Desenvolvimento Integrado (*Integrated Development Environment* ou IDE) e Axis (AXIS, 2004), Protocolo de comunicação via *Web Services*;
- PostgreSQL (POSTGRESQL, 2005): sistema de gerenciamento de banco de dados relacional e orientado a objetos;
- LDAP (*Lightweight Directory Access Protocol*) (LDAP, 2004), OpenLDAP (OPENLDAP, 2004) e JNDI (*Java Naming and Directory Interface*) (JNDI, 2004) que são respectivamente: protocolo para acessar servidores de diretórios; implementação *open source* do protocolo LDAP e API (*Application Programming Interface*) J2EE (J2EE, 2004) para manipular objetos e servidores LDAP;
- *Access to Biological Collection Data* ou ABCD (ABCD-a, 2004). Padrão internacional para catalogação de espécies de biodiversidade.
- Darwin Core (DARWINCORE, 2004). Padrão internacional para catalogação de espécies de biodiversidade.
- MapServer (MAPSERVER, 2005), estudo preliminar para criação de mapas de regiões específicas do globo terrestre, mostrando a distribuição mais provável das espécies de bromélias (de acordo com a obtenção prévia de dados, características físicas e pontos geográficos) na natureza.

2.1 Métodos de Desenvolvimento

2.1.1 Especificação

2.1.1.1 UML (*Unified Modelling Language*)

A *Unified Modeling Language* ou UML (RUMBAUGH, 2003) é a padronização de desenvolvimento orientado a objetos para especificação, documentação, visualização e desenvolvimento de sistemas. Através dos diagramas da UML é possível representar sistemas de software sob diversas perspectivas de visualização. Isto facilita a comunicação entre todas as pessoas envolvidas no processo de desenvolvimento do sistema, uma vez que os diagramas possuem um vocabulário de fácil entendimento.

Além disso, a UML permite que diferentes grupos de desenvolvimento interpretem de maneira correta e sem ambigüidades os modelos gerados por outros desenvolvedores, contribuindo para a continuidade do projeto ao mesmo tempo em que é minimizada a ocorrência de erros.

Dessa forma, a modelagem do sistema através da UML tem os seguintes passos:

i. Diagramas de Casos de Usos

Descreve a funcionalidade do sistema na visão do usuário.

ii. Diagrama de Classes

Descreve a estrutura do sistema através das classes.

iii. Diagrama de Objetos

Descreve o sistema através dos objetos que estão instanciados em um determinado instante.

iv. Diagrama de Componentes

Descreve a estrutura física da implementação do sistema.

v. Diagrama de Implantação

Descreve a topologia da estrutura de hardware, pode descrever também os elementos de software alocados nos elementos de hardware.

2.1.2 Projeto

2.1.2.1 XP (*Extreme Programming*)

Extreme Programming é uma disciplina de desenvolvimento de software baseado em valores de simplicidade, comunicação, feedback e coragem. Ela faz com que toda a equipe fique envolvida através de pequenas práticas, com o feedback necessário para a equipe se interar e direcionar estas práticas para um único objetivo.

Em *Extreme Programming*, cada colaborador do projeto é parte integral da 'equipe toda'. A equipe se forma através de um representante chamando 'cliente' que se senta com a equipe e trabalha com eles diariamente. No caso do projeto o cliente foi representado pelo orientador e a equipe, pelos orientados, composto por três pessoas. A cada duas semanas foram feitas reuniões para planejar novamente e situar outros grupos de trabalho sobre o andamento da pesquisa.

As equipes de XP usam uma forma simples de planejamento e investigação para decidir o que será feito a seguir e prever quando o projeto será feito. A equipe produz o software em uma série de pequenos módulos que passam por todos os testes que o 'cliente' tenha definido. Estes módulos são construídos de tal maneira que o sistema esteja sempre integrado e executável sempre.

2.1.2.2 Computação Orientada a Serviços

A Computação Orientada a Serviços ou *Services Oriented Computing* (SOC) (SOA, 2003) é um paradigma computacional que utiliza serviços como os elementos principais para

o desenvolvimento de aplicações. Os Serviços são oferecidos por Provedores, ou seja, organizações que possuem a implementação e a descrição dos serviços. Logo, os Serviços que são oferecidos por diferentes comunidades e empresas por toda a Internet, permitem uma infra-estrutura de computação distribuída para várias aplicações.

A Computação Orientada a Serviços envolve camadas de serviços, funcionalidade e regras que é descrita através da Arquitetura Orientada a Serviços ou SOA (SOA, 2003), de acordo com a Figura 1. Os serviços básicos, sua descrição e as operações básicas constituem a base do SOA. As camadas mais altas do SOA fornecem suporte adicional para a composição e gerenciamento dos serviços.

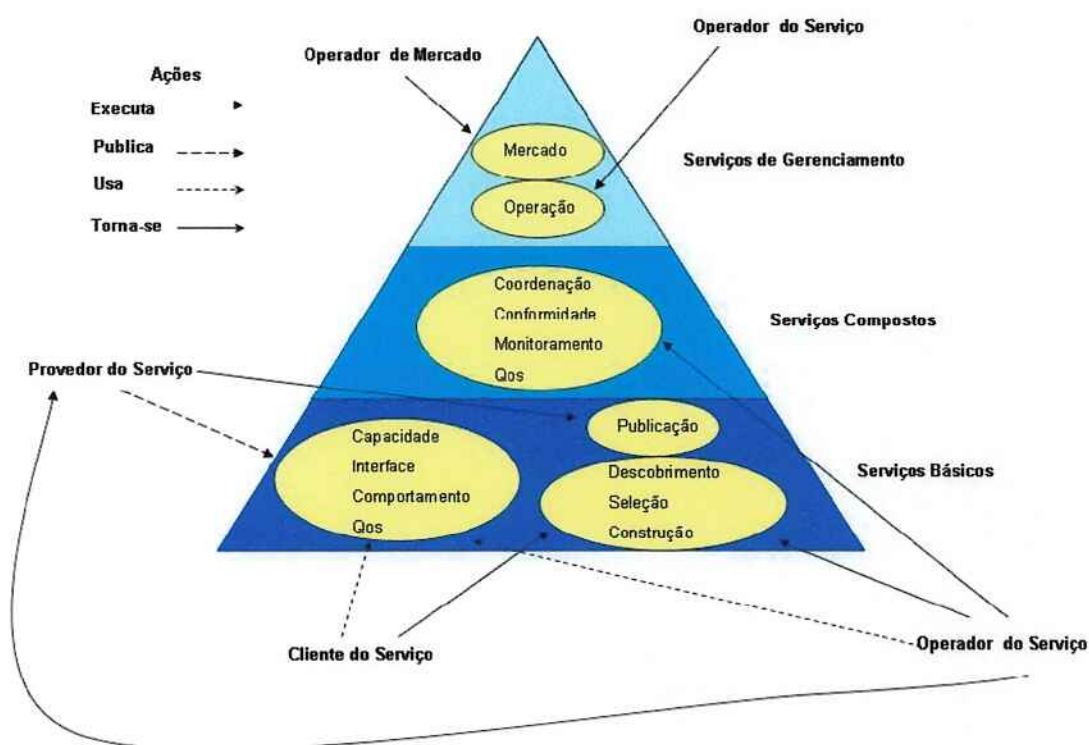


Figura 1: Camadas de Serviços.

A camada Serviços Compostos contém as regras e as funcionalidades para a consolidação de múltiplos serviços em um único serviço composto. O resultado deste serviço composto pode ser utilizado pelo Agregador de Serviços como componentes que podem ser utilizados como aplicações ou soluções por clientes. Agregadores de Serviços então transformam-se em Provedores de Serviços que publicam a descrição dos serviços.

Os Agregadores de Serviços desenvolvem especificações ou códigos que permitem que a camada Serviços Compostos tenha as seguintes funções:

- **Coordenação:** Controle de execução dos serviços e do gerenciamento de transações.
- **Monitoramento:** Monitora os eventos e as informações produzidas pelos serviços.
- **Conformidade:** Assegura a integridade do serviço através dos tipos de parâmetros, além de promover a fusão de informações.
- **Qualidade de Serviço:** Inclui ao serviço o custo, desempenho, segurança, autenticação, privacidade, integridade, escalabilidade e a disponibilidade.

Para o gerenciamento de aplicações críticas, o SOA (SOA, 2003) possui serviços de gerenciamento na camada de gerenciamento. Em particular, as operações na camada Gerenciamento de Serviços do SOA tem como objetivo o suporte para aplicações críticas que necessitam de empresas para gerenciar a plataforma de Serviço. A organização responsável pela operação de gerenciamento é conhecida como Operador de Serviço, que pode ser um cliente ou um Agregador de Serviço, dependendo dos requerimentos da aplicação.

A aplicação dos conceitos de Computação Orientada a Serviços na WEB é denominada *Web Services*. A *Web Services* é identificado por um URI (*Universal Resource Identifier*) no qual a descrição dos Serviços e a sua transmissão utilizam os padrões da Internet.

A *Web Services* tem sido envolvida como um meio de integrar processos e aplicações enquanto as semânticas e protocolos das transações tradicionais têm se mostrado inapropriado devido aos seus longos períodos de execução.

2.1.2.3 **web services**

Essa arquitetura é baseada em conceitos de *Middlewares* Orientados a Objetos e Computação Orientada a Serviços, que foram aplicados a Internet. Os elementos que compõem a Arquitetura de *Web Services* são discutidos a seguir:

i. SOAP (*Simple Object Access Protocol*) (W3c-b, 2005) é um mecanismo que permite definir mensagens de requisições e respostas (*request/response*), utilizando principalmente o protocolo HTTP (W3c-d, 2005) para transmitir mensagens. SOAP prevê também outros protocolos, tais como: FTP, SMTP ou Telnet (W3c-d, 2005). O protocolo SOAP define mensagens em XML (XML, 2005), que descreve um envelope que contém informações de localização e contexto de serviço, e um corpo com o serviço e os parâmetros requisitados; no caso de respostas o corpo contém os valores retornados. O protocolo SOAP é responsável por garantir a interoperabilidade para a transferência de informações entre ambientes computacionais distintos.

ii. WSDL (*Web Services Description Language*) (W3c-c, 2004) linguagem baseada em XML, utilizada para descrever os serviços, parâmetros e a localização na WEB, utilizando um modelo de descrição neutro em relação a uma linguagem específica de implementação de um provedor do serviço ou de um cliente.

ii. UDDI (*Universal, Discovery, Description and Integration*). (IBM, 2005) é um padrão de *broker* para a Internet, que vem sendo especificado por um consórcio de aproximadamente 300 empresas, lideradas pela IBM, Microsoft e HP. De acordo com a Figura 2, pode-se observar que este *broker* central permite que provedores registrem seus serviços bem como os clientes localizem serviços. Os registros são acessados pelos clientes através de SOAP, e obtém uma descrição dos serviços em WSDL, gerando cópias das informações ou *proxy* do lado do cliente que invoca o serviço WEB.

Através da Arquitetura da *Web Services* é possível o desenvolvimento de sistemas distribuídos agregando serviços e informações disponíveis em outros sistemas dinamicamente, permitindo a construção de portais e desenvolvimento de aplicações que envolvam a integração de negócios e, em especial, aplicável para a troca de informações entre bases de biodiversidade. Porém deve-se observar que essa tecnologia é imatura e vem sendo rapidamente evoluída.

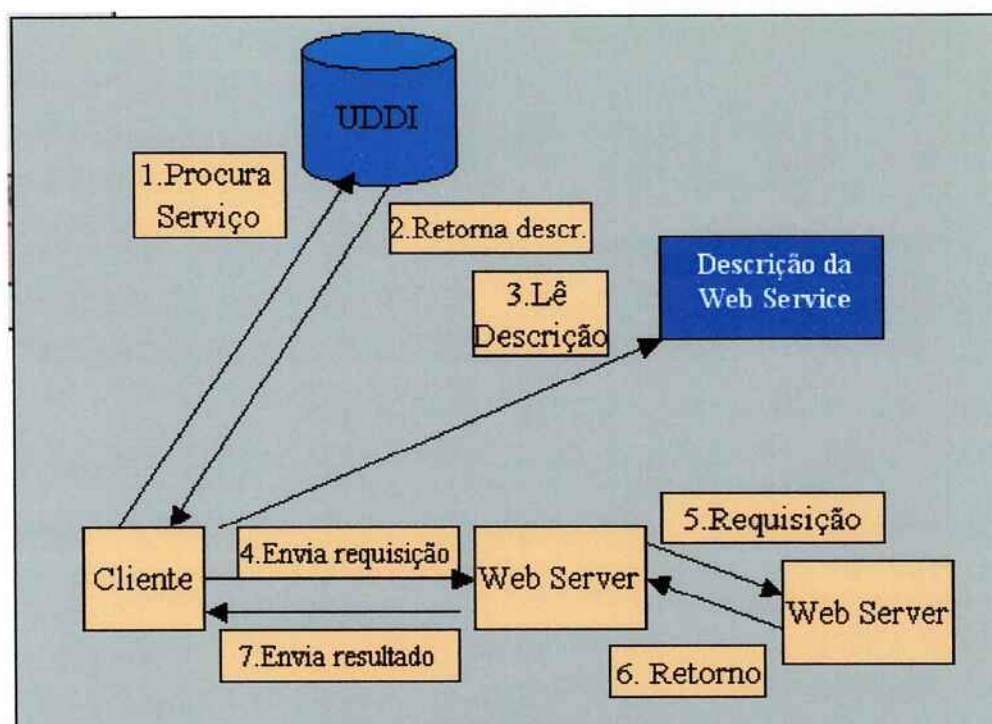


Figura 2: Arquitetura utilizada por *Web Services*.

De acordo com a Figura 2, pode-se observar que o Sistema é baseado na Arquitetura de *Web Services*, definindo um Portal que é responsável por interagir com os usuários, localizar Provedores e serviços disponíveis utilizando o UDDI, e formular as requisições para os Provedores. O Provedor recebe as requisições, faz a tradução do serviço de consulta requisitado numa consulta válida para o banco de dados local, executa a consulta, monta a mensagem de resposta com o resultado obtido e envia para o Portal.

2.1.2.4 DiGIR

O projeto DiGIR surgiu com o principal intuito de evitar múltiplos desenvolvimentos de sistemas inconsistentes entre si, visando juntar esses esforços em um único projeto capaz de criar uma comunidade de especialistas de tal forma a resolver o problema de integração de sistemas de bancos de dados distribuídos.

Dentre os principais objetivos, estão a definição de um protocolo para consultar dados estruturados de banco de dados heterogêneos via Internet e implementação deste através de um Provedor (*Provider*) e um portal (*Portal Engine*) que utilizem este protocolo. Também há três preocupações: utilizar protocolos abertos e padronizados, como por exemplo HTTP

(HTTP, 2004), XML e UDDI; facilitar a instalação de um novo provedor de dados; e por último, desenvolver esse sistema de modo *opensource*.

A arquitetura DiGIR tem quatro elementos importantes, a saber: a interface com o usuário, o protocolo, o Portal e o Provedor.

De um modo geral, os serviços de um Provedor são cadastrados no *Registry*, que guarda os locais de outros Provedores. O Portal é o responsável por acessar o *Registry* e descobrir os Provedores disponíveis. O usuário, através da sua interface final, pode fazer sua consulta ao Provedor através do Portal, que consulta o Provedor relativo à consulta efetuada pelo usuário. O Provedor, mediante requisição, entrega as informações pedidas ao Portal que por sua vez, as repassa para a interface.

O protocolo utilizado entre Provedor – Portal, Portal – Interface define formatos de mensagens de resposta e requisição, que são de três tipos: *metadata* (*Metadata*), pesquisa (*Search*) e inventário (*Inventory*). Estas mensagens não são afetadas pela estrutura dos dados que elas transferem, porém os Provedores devem estar conforme a um *federation schema*.

O Portal tem como principais funções: realizar consulta ao *Registry* de possíveis Provedores; determinar através de *metadata* de Provedores quais Provedores devem ser consultados; enviar requisições a múltiplos Provedores; organizar as respostas dos Provedores; retornar resultados para o usuário; e atividades de logs.

O Portal está baseado em ferramentas Apache (Apache, 2005), JDK 1.4 (SUN's *Java Development Kit*) e Tomcat, além do software do Portal (Portal software). Este software permite configurar o *Engine* de tal forma a procurar o Provedor ao qual pode se comunicar; configurar e customizar a parte de apresentação ao usuário; além de possuir os scripts para iniciar e parar o Portal (*Engine start e stop scripts*), assim como o *Portal Presentation* (*Presentation start e stop scripts*).

O Provedor é o responsável por receber as requisições, coletar informações do banco de dados, enviar os resultados ao usuário, informar *metadata* de tal modo a descrever qual o tipo de dados e a disponibilidade e por fim, fazer o *log* das requisições. Ele deve estar conforme com algum dos *federation schema* conhecido.

Originalmente, o Provedor está baseado em ferramentas como Apache e PHP (PHP, 2005), além do software para o Provedor (*Provider software*) desenvolvido pelo projeto

DiGIR. Este software permite definir o *metadata* do Provedor, fazer a conexão com o banco de dados, estabelecer tabelas de relacionamento para poder mapear dados das tabelas de acordo com o padrão taxonômico DarwinCore (DARWINCORE, 2005). Este padrão é o que possibilita a integração de banco de dados distintos de Biodiversidade

O *Registry* tem como característica divulgar os Provedores cadastrados, rever, caso o Portal tenha apenas um Provedor como padrão.

Já a interface com o usuário (*User Interface*) deve ser capaz de organizar e enviar os pedidos ao Portal; e receber e interpretar as respostas do Portal.

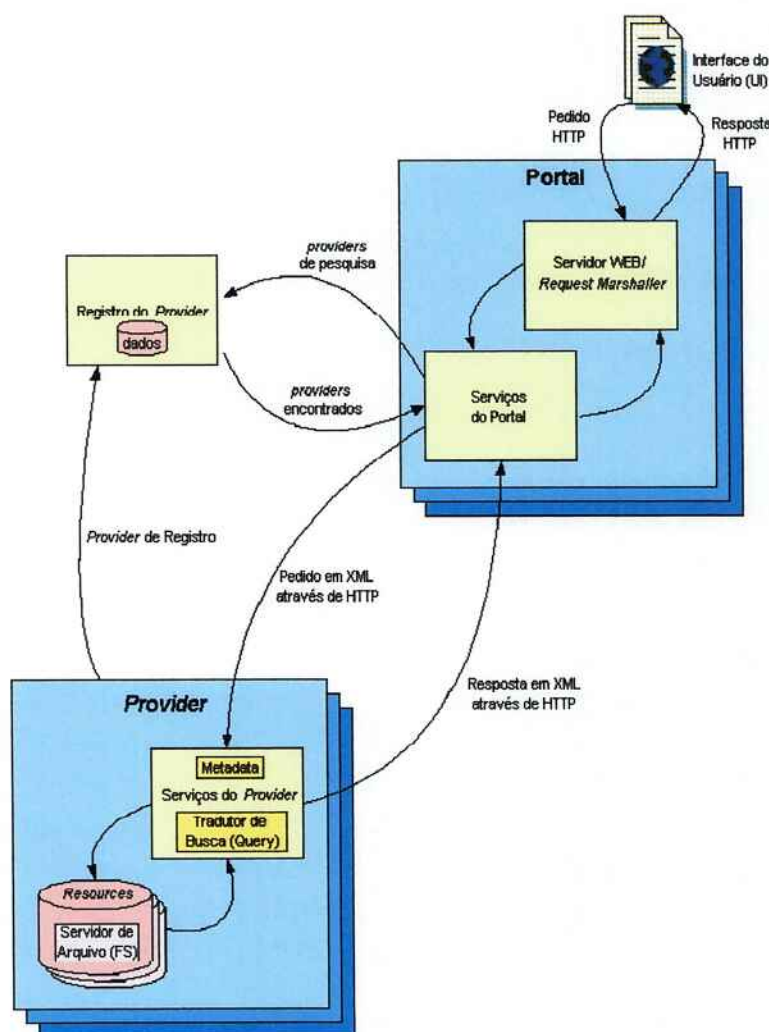


Figura 3: Arquitetura do DiGIR

2.2 Ferramentas de Desenvolvimento

2.2.1 Linguagens de programação

2.2.1.1 J2EE

J2EE é uma arquitetura para desenvolvimento de aplicações corporativas baseada na criação de aplicações na Internet.

2.2.1.2 JSP (Java Server Page)

A linguagem JSP é um dos componentes da arquitetura J2EE utilizada para a criação de páginas dinâmicas na web. Páginas dinâmicas são aquelas que a cada vez que são solicitadas podem mostrar dados diferentes (de acordo com a solicitação do próprio usuário). Com JSP, criadores de páginas da web e programadores podem rapidamente incorporar elementos dinâmicos dessas páginas, utilizando Java embutido e algumas *tags* de marcação simples. Estas *tags* fornecem ao criador de HTML (*Hyper Text Markup Language*) (HTML, 2004) um meio de acessar dados e lógica de negócio em objetos Java sem a necessidade de conhecer as complexidades do desenvolvimento de aplicações.

2.2.2 Ambiente de Desenvolvimento

2.2.2.1 Eclipse

O Eclipse é uma ferramenta cuja funcionalidade não é somente para o desenvolvimento de códigos java, mas também para visualização de base de dados, edição de arquivos XML, JSP, HTML, JavaScripts (JAVASCRIPTS, 2004) entre outras funcionalidades. Tem como principal atrativo a personalização do ambiente de desenvolvimento através dos *plugins* (arquivos que contém informações para alterar, melhorar ou estender operações de uma família de programa de aplicativos) que são desenvolvidos pela comunidade *opensource*.

2.2.2.2 Axis

Axis é uma implementação do SOAP (*Simple Object Access Protocol*) do Apache; trata-se de um mecanismo que permite definir mensagens de requisições e respostas, utilizando principalmente o protocolo HTTP, para transmitir informações e garantir a interoperabilidade para a transferência de dados entre ambientes distintos. Através de um *plugin*, é possível integrá-lo ao Eclipse.

2.2.2.3 CVS

O CVS (CVS, 2004) (*Concurrent Versions System*) é um software que auxilia no controle de versões. Seu uso permite que diversas versões de um projeto sejam armazenadas em um diretório comum (o repositório), armazenando apenas as diferenças entre as versões e os arquivos comuns, permitindo assim gerenciamento eficiente dos arquivos gerados. O CVS permite também que diversos desenvolvedores trabalhem em um mesmo projeto de forma harmônica, sendo que cada um recebe sua área de trabalho e o gerenciamento das alterações é feito de forma a evitar conflitos.

2.2.2.4 Tomcat

Tomcat é um *Servlet Container*, ou seja, é um servidor onde são instaladas *Servlets* para tratar as requisições que o servidor receber. É neste servidor que as aplicações JSP serão armazenadas e executadas.

2.2.2.5 PostgreSQL

O PostgreSQL (POSTGRESQL, 2005) é um sistema de gerenciamento de banco de dados relacional estendida, suportando quase todas as construções SQL, incluindo subseleções, transações, tipos definidos pelo usuário e funções. Atualmente está disponível seu código fonte, além de binários pré-compilados em diversas plataformas.

É possível utilizar o PostgreSQL em vários sistemas operacionais, dentre os quais o Windows (WINDOWS, 2005), Linux (LINUX, 2005), Unix (UNIX, 2005) ou em qualquer sistema compatível com as especificações POSIX (*Portable Operating System Interface*) (POSIX, 2005). O PostgreSQL (POSTGRESQL, 2005) permite a criação de uma base de dados de grandes proporções. Cada tabela pode ter até 16 *Terabyte*, sendo que cada linha pode ter até 1,6 *Terabyte* e cada campo 1 *Gigabyte*.

2.2.2.6 MapServer

O MapServer é um ambiente de desenvolvimento *OpenSource* para construção de aplicações na Internet especialmente ativas. Foi originalmente desenvolvido pela *University of Minnesota* (UMN) ForNet, projeto em cooperação com a NASA e com o *Minnesota Department of Natural Resources* (MNDNR). Alguns aprimoramentos foram feitos pelo MNDNR e pelo *Minnesota Land Management Information Center* (LMIC). O atual desenvolvimento é financiado pelo projeto TerraSIP, um projeto patrocinado pela NASA entre o UMN e o *consortium of land management interests*.

O sistema suporta o "MapScript", o que permite o uso de linguagens script muito populares como Perl, Python e até Java para acessar a API C do MapServer. O MapScript proporciona um rico ambiente para desenvolvimento de aplicações que integram dados muito diferentes. Se o dado possui um componente espacial e deve ser pego em outro ambiente "scripting" em particular, um mapeamento entre os sistemas será realizado para permitir tal transação. Por exemplo, usando o módulo JAVA é possível integrar informações de qualquer banco de dados com dados GIS - *Geographic Information Systems* em um único mapa gráfico ou em uma página da WEB.

O MapServer não é um sistema GIS completo, porém fornece funcionalidades suficientes para sustentar uma ampla variedade de aplicações web. Além da leitura de dados GIS, ele permite a criação de "imagens de mapas geográficos", isto é, mapas que conseguem mostrar o conteúdo claramente para os usuários.

Contudo, o MapServer não será utilizado nessa implementação do estudo de caso, devido a incompatibilidade de escalas entre mapas de diferentes organizações Brasileiras,

mostrando que sem uma devida padronização de escalas de mapas, não poderemos realizar o cruzamento de informações entre, perdendo assim valiosas informações de biodiversidade, como por exemplo, o efeito do relevo e do clima na ocorrência de algumas espécies de bromélias.

2.2.3 Protocolo de Segurança

2.2.3.1 LDAP

LDAP (*Lightweight Directory Access Protocol*) é um protocolo usado para acessar servidores de diretórios. Um diretório é um tipo especial de banco de dados que armazena informações em uma estrutura de árvores.

O conceito é similar a uma estrutura de diretórios, exceto neste contexto, o diretório raiz ou *root*, é o “o mundo” e o primeiro nível de subdiretórios são os países. Níveis abaixo da estrutura de diretório contêm entradas de empresas, organizações ou lugares, e indo mais abaixo, encontra-se dados de pessoas, equipamentos ou documentos.

Para se referir a um arquivo em um subdiretório, usa-se o seguinte ou semelhante a:

```
/usr/local/myapp/docs.
```

A barra indica cada divisão na referência e a sequência é lida da esquerda para direita.

O equivalente em LDAP seria o nome distinto (*distinguished name*), simplesmente referido como *dn*. Um exemplo de *dn* seria:

```
cn=Maria, ou=secretaria, o=empresa X, c=br
```

A vírgula divide cada pedaço da referência e a sequência é lida da direita para a esquerda. Pode-se ler este *dn* como:

```
País (country) = br
Organização (organization) = empresa X
Unidade Organizacional (organizationalUnit) = secretaria
Nome (commonName) = Maria
```

Assim como não existem regras sobre como se deve organizar a estrutura de diretórios, o gerente do servidor de diretórios pode configurar qualquer estrutura que tenha um significado para algum propósito, desde que se use algumas convenções nessa estrutura.

A seguir, serão apresentadas algumas ferramentas que implementam o modelo de diretórios LDAP.

2.2.3.2 Open LDAP

OpenLDAP (OPENLDAP, 2005) foi desenvolvido a partir do LDAP 3.3 da Universidade de Michigan. Seu projeto é gerenciado pela *OpenLDAP Foundation* (OPENLDAP, 2005). O seu projeto inclui: *slapd* (servidor LDAP *stand-alone*); *slurp* (*stand-alone LDAP replication server*); bibliotecas que implementam o protocolo LDAP, utilitários, ferramentas e exemplos de clientes.

O projeto OpenLDAP é um esforço em conjunto para desenvolver um conjunto LDAP *opensource* de aplicações e ferramentas de desenvolvimento. O projeto é gerenciado por uma comunidade global de voluntários que usam a Internet para comunicar, planejar e desenvolver o conjunto OpenLDAP e sua respectiva documentação.

2.2.3.3 JNDI

JNDI (*Java Naming and Directory Interface*) (JNDI, 2005) é a API J2EE que fornece uma interface padrão para localizar usuários, máquinas, objetos, redes e serviços. Nesse contexto, permite acessar serviços de nomes e diretórios através de aplicações Java. Como será utilizado o LDAP para fazer a autenticação no sistema, essa API é indispensável para retornar os dados guardados dentro dos diretórios LDAP. Outros métodos como pesquisa, alteração, manipulação e busca em diretórios e interação com LDAP pode ser feitos através do JNDI.

2.3 Padrões Internacionais

Atualmente, existem diversos padrões para catalogação de espécies que vem sendo desenvolvidos. Contudo, muitos são direcionados para uma determinada aplicação, tornando-os restritos a aquele sistema ou espécie. Entretanto, existem dois padrões que tem ganhado adeptos, devido principalmente a sua generalidade na catalogação de espécies, que são os descritos a seguir:

2.3.1 *ABCD - Access to Biological Collection Data.*

O padrão *Access to Biological Collection Data* (ABCD-a, 2005) ou ABCD é um padrão internacional desenvolvido por taxonomistas da TDWG (*International Working Group on Taxinomic Databases*) (TDWG, 2004), sendo que esta é uma organização internacional que possui como objetivo a padronização e a integração de base de dados taxonômicos de biodiversidade.

O ABCD consiste em um conjunto geral de informações para catalogação de diferentes espécies em biodiversidade. Isto permite uma padronização na catalogação de espécies, ao mesmo que o ABCD permite, devido a sua grande abrangência, a possibilidade de se cadastrar as mais diferentes espécies.

A representação adotada pelo ABCD (ABCD-b, 2005) foi através da linguagem XML (XML, 2004), através de um esquema. Este esquema permite uma maior integração entre sistemas diferentes, já que o XML é suportado por diferentes plataformas, tornando assim o ABCD portátil e de fácil entendimento.

2.3.2 *Darwin Core*

O DarwinCore (DARWINCORE, 2005) é um conjunto das informações comumente encontradas na maioria das coleções biológicas existentes. Este padrão pressupõe que apenas as informações mais importantes de cada espécie sejam catalogadas. Logo, o DarwinCore apresenta e especifica essas informações de forma a criar um padrão mínimo de informações que os bancos de dados e os sistemas de consulta devem tratar.

Dessa forma, todos os campos do padrão DarwinCore possui um campo equivalente no padrão ABCD, principalmente pela grande abrangência desta última.

2.4 Conclusão

No desenvolvimento do Sistema de Informação de Bromélias foi utilizado o padrão internacional de catalogação de espécies ABCD para a elaboração da base de dados sendo o Gerenciador de Banco de Dados utilizado o PostgreSQL. Para a segurança, foi utilizado o protocolo de LDAP. Todo o sistema foi baseado nas premissas da Computação Orientada a Serviços, sendo que para sua modelagem, foram utilizadas as diretrizes de *Extreme Programming* e UML. Para a implementação de toda a arquitetura, o IDE Eclipse promoveu um ambiente de desenvolvimento integrado, permitindo a criação de um ambiente com todas as ferramentas necessárias para o desenvolvimento.

Inicialmente foi definido ambiente de desenvolvimento MapServer para o tratamento de imagens de satélites. Contudo, devido a incompatibilidade de escalas de mapas de diferentes organizações foram discutidas outras opções, sendo a escolhida a implementação de um Provedor de Mapas, com transferência via *WebServices*.

3. ARQUITETURA DO SISTEMA

Neste capítulo são discutidos os requisitos de sistemas de informação considerados, além das soluções de arquitetura de sistemas adotadas. De maneira geral a especificação da Arquitetura do sistema foi baseada nos seguintes princípios:

- Computação Orientada a Serviços, utilizando *Web Services*;
- Base de dados distribuídas integrando informações disponibilizadas pelos pesquisadores em diferentes lugares, permitindo um compartilhamento de informações, baseadas no padrão ABCD através de Provedores.
- *Metadata* baseada no padrão de catalogação de espécies ABCD, possibilitando uma troca homogênea de informações entre os Provedores.

3.1 sistema de informação

A aplicação dos conceitos de *Web Services* e base dados distribuídas permitem obter uma arquitetura em que as consultas de informações são feitas através de um Portal que faz a requisição do Serviço em todas as bases de dados disponíveis. Seguindo esse conceito, todo o sistema foi baseado na seguinte Arquitetura básica:

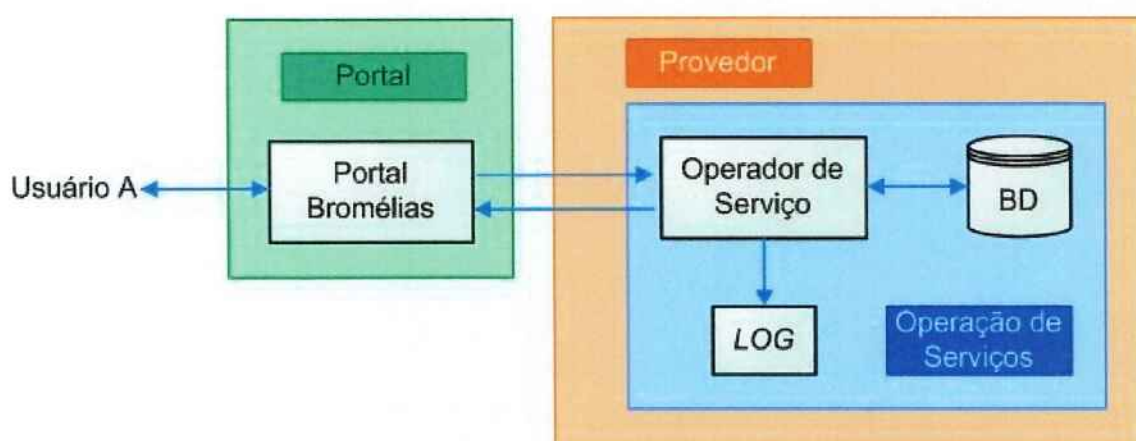


Figura 4: Arquitetura Básica

A arquitetura proposta na figura acima separa o sistema em instituições, cada uma contendo seu Provedor, sua base de dados e seu portal. Cada Instituição possui um portal próprio, no qual é feito não só as consultas aos Serviços de sua base de dados, como também de outras instituições. Toda essa troca de informações é controlada por perfis de acesso, no

qual o usuário se autentica no sistema de informação, obtendo assim um perfil associado a sua autenticação. Dessa forma, baseado em seu perfil de acesso, cada instituição pode autorizar o usuário a acessar determinados serviços disponibilizados pelo Provedor. Isso permite uma arquitetura distribuída, na qual pesquisadores colecionadores e formuladores de políticas de conservação possam ter sua própria base de dados e compartilhar essas informações de acordo com a *metadata* disponibilizada através do padrão ABCD (ABCD-a, 2005).

A arquitetura baseada em serviços mostra vários Portais em diferentes Instituições, na qual estes realizam requisições de serviços aos diversos Provedores disponíveis. Primeiramente, o Portal faz a requisição do serviço para todos os Provedores disponíveis, recebendo assim um conjunto de informação fornecida pelos diversos Provedores que gerenciam as bases de dados distribuídas em diferentes localidades. Logo, o Portal que fez a requisição, após o recebimento dos dados, faz toda a manipulação dos dados recebidos dos diferentes Provedores permitindo assim um primeiro processamento dos dados, tornando os dados legíveis ao usuário final do sistema, ou seja, o Portal faz a compilação de todos os dados, mostrando ao usuário final apenas o resultado da consulta.

A possibilidade de ter nesta Arquitetura base de dados distribuídas permite que cada pesquisador possa ter sua própria base de dados, descentralizando a informação, além de facilitar sua implementação, uma vez que o pesquisador tendo grande quantidade de informações, elimina a dificuldade que seria exportar todas a informação sobre biodiversidade para uma base de dados remota. Além disso, uma base local, permite ao pesquisador um maior controle sobre sua informação, ou seja, ele saberá quem e quando suas informações estão sendo acessadas, possuindo assim um maior controle a informação disponibilizada.

Os Provedores disponibilizam através de serviços definidos localmente, todas as operações disponíveis sobre a base de dados local relacionada com este Provedor. Dessa maneira, os Provedores têm a função de interface entre o banco de dados e os Portais, realizando a operações (consulta, inserção, alteração e remoção) na base de dados, além de disponibilizar serviços aos Portais para a geração de das ações requisitadas pelos usuários dos Portais.

A arquitetura também contempla o Provedor de mapas de imagens, permitindo o pesquisador obterem serviços de transferência de imagens entre o Provedor e o Portal, tornando assim a pesquisa pelos usuários mais completa

3.2 sistema de segurança

Baseado na arquitetura do Sistema de Informação, o sistema de segurança proposto busca um meio de permitir o controle de acesso aos dados de biodiversidade e este acesso é concedido apenas para serviços de acordo com a origem da requisição, o perfil associado e a identificação do usuário que fez a requisição. Diante disso, levantou-se a seguinte arquitetura lógica. (Figura 5)

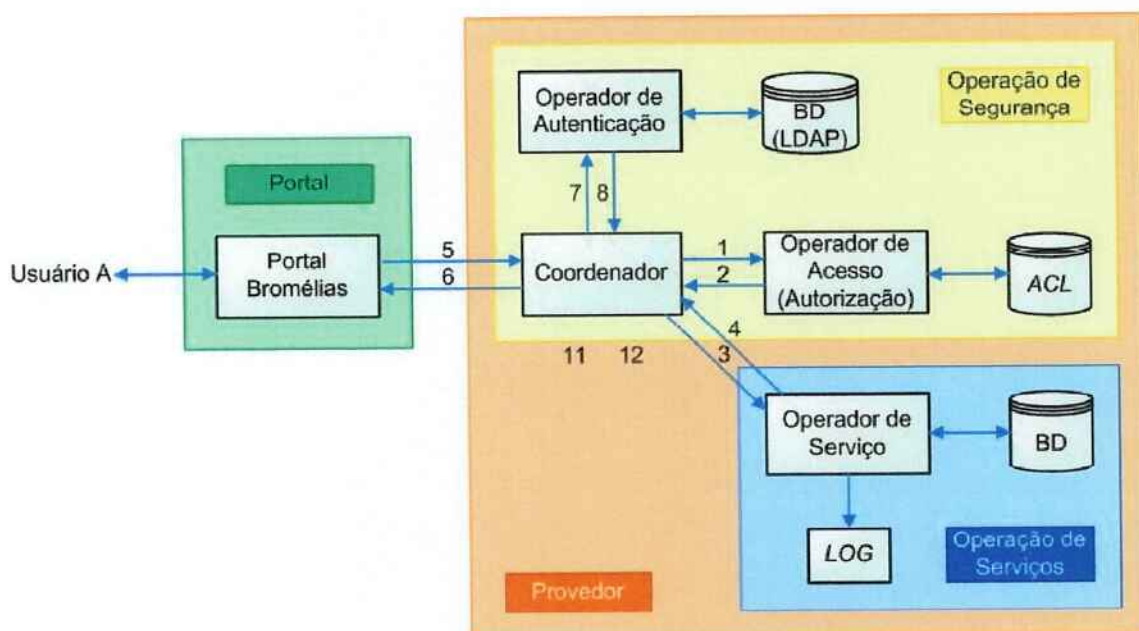


Figura 5: Arquitetura Segurança

3.2.1 *Estudo de Caso: Usuário A acessando serviços a partir do Portal de Bromélias*

Descrevendo com detalhes o processo na Figura 5: o usuário A acessa o portal do sistema, como por exemplo o Sistema de Informações de Bromélias. O portal de Bromélias através de *login*, senha e local digitados pelo usuário A envia estes dados para o Coordenador (passo 5) e caso o usuário não seja portador daqueles dados, de acordo com o Coordenador (passo 6), é permitido a entrada como convidado, não podendo acessar conteúdo que necessite uma autorização mais alta.

Detalhando os passos anteriores: o Coordenador, ao receber *login*, senha e local do Portal (passo 5), envia esses dados para Operador de Autenticação (passo 7), que se encarrega de verificá-los em uma base de dados, por exemplo LDAP, e retornar o perfil e local do usuário A (passo 8) para que o Coordenador possa responder ao portal e este termina por autenticar o usuário A (passo 6).

Após isso, cada vez que o usuário A solicitar um serviço da base de dados, o portal entrega além do pedido da transação, a autenticação do usuário (passo 5), que será enviada ao Operador de Acesso (Autorização) através do Coordenador (passo 1). De posse da autenticação e do serviço do usuário, o Operador de Acesso valida a autorização (passo 2) para que o Coordenador entregue o pedido do serviço para o Operador de Serviço (passo 3) e este finalmente entrega os dados do serviço desejado (passo 4) para o Coordenador entregar ao Portal (passo 6).

3.2.2 *Estudo de Caso: Comunicação entre entidades dentro de uma comunidade*

Mais especificamente, esta arquitetura permite também que outras entidades possam se integrar (figura 6). Isto acaba sendo de grande importância, pois permite que entidades de provedores possam se comunicar, com o objetivo de fornecer informações relacionadas ao domínio de aplicação, como por exemplo uma comunidade de provedores de informações sobre bromélias.

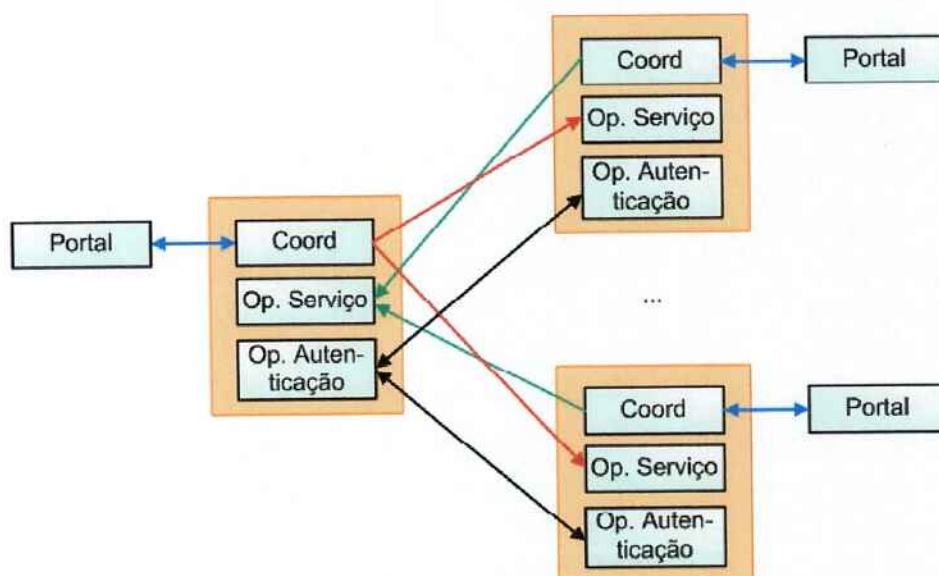


Figura 6: Comunicação entre entidades

Dessa maneira, uma vez que os sistemas de autenticação são integrados (setas pretas), um coordenador pode ter acesso a outros operadores de serviços, onde estão as bases de dados, (setas vermelhas), assim como outros coordenadores que fazem parte da comunidade (setas verdes).

3.2.3 *Estudo de Caso: Autenticação entre entidades dentro de uma comunidade*

Também é possível, de acordo com a figura 7, permitir que a parte responsável pela autenticação (por exemplo usando serviço LDAP), caso não encontre a autenticação localmente, tente fazer a autenticação em outros locais. Assim, permite que o responsável da base de dados gerencie seu acesso, mas também permita acesso de outros portais, desde que tenham um compromisso um com outro de permitir acesso a certas informações de seus operadores de serviços.

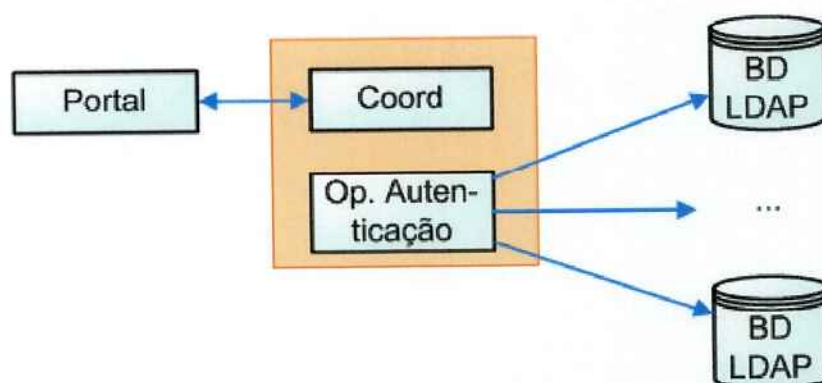


Figura 7: Autenticação entre entidades

3.2.4 *Estudo de Caso: Serviços Compostos*

Do mesmo modo, a arquitetura lógica proposta permite que serviços compostos sejam feitos: (Figura 8). Assim, para fornecer um serviço composto, o operador de serviço é o responsável por fazer requisição a outro coordenador, fazendo autenticação e autorização para receber dados necessários para completar o serviço requisitado.

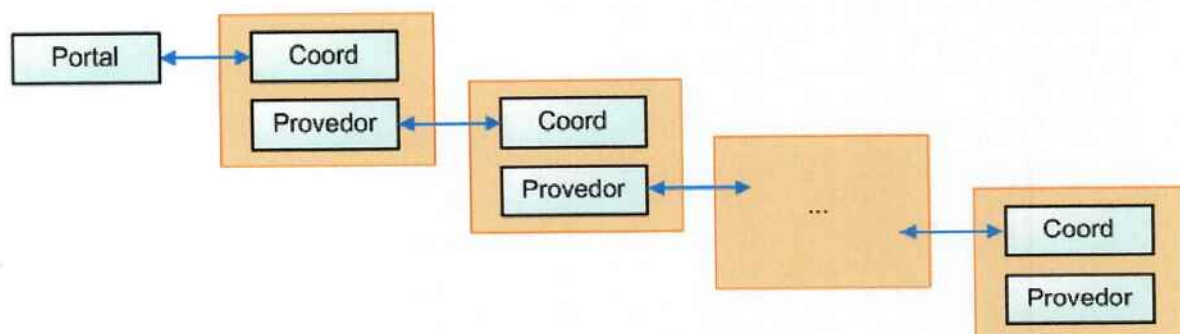


Figura 8: Serviços compostos

3.3 sistema de Informação – Arquitetura SOA

Neste item é discutido o mapeamento dos serviços considerados na arquitetura SOA (*Service Oriented Architecture*). O SOA, na realidade representa sistemas independentes que expõem serviços e são de propriedade de diferentes organizações, especialmente os Provedores. Os processos de autenticação e autorização considerados no sistema, têm como papel principal a troca de informação entre as diferentes organizações de pesquisa de Bromélia, transformando assim informações isoladas em conhecimento inter-relacionado. A Figura 9 representa os serviços que existem no Sistema de Informação. É basicamente constituído por um portal de bromélias, que contém as interfaces que permite ao usuário acessar um serviço específico de bromélia; um Provedor de Mapas que é um repositório de mapas; um Provedor de Bromélias que é responsável pela autenticação e autorização dos usuários. Note que este também pode fazer requisições para outros Provedores para obter informações.

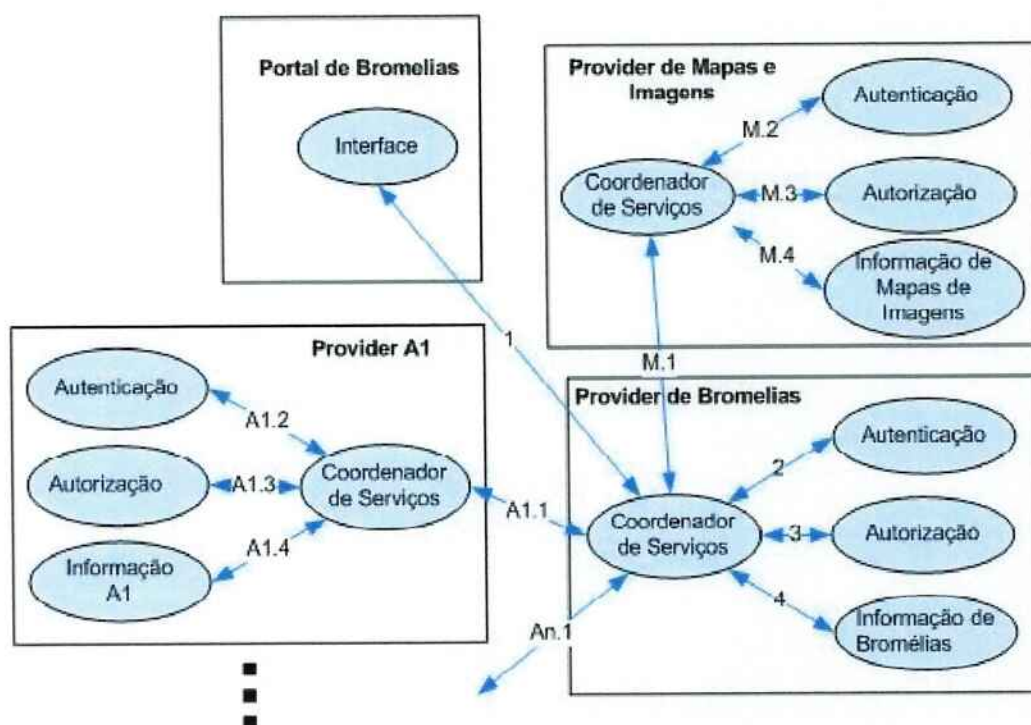


Figura 9: Definição de Serviços no Sistema

A Tabela 1 mostra as camadas e as funcionalidades descritas na arquitetura SOA, e os diferentes Serviços ilustrados na figura 9.

Tabela 1: Mapeamento das funcionalidades da Arquitetura SOA

Camada de Serviço	Funcionalidade	Representado na arquitetura por
Composição de Serviços	Coordenação, Conformidade e Monitoramento	Coordenador de Serviços
Serviços Básicos	Capacidade, Interface, Comportamento, QoS, Publicação, Descobrimto, Seleção.	Autenticação, Autorização, Informação de Bromélias, Informação de Mapas, Informação A1 e Portal de Bromélias

De acordo com a figura 9, o processo se inicia com o Portal de Bromélias requisitando um serviço para o Provedor de Bromélias (Passo 1). Antes de o Provedor processar a informação requisitada pelo Portal, é necessária a verificação de autenticação (Passo 2) e autorização (Passo 3). Após essa verificação é que o Provedor permite o acesso à informação de Bromélias no Banco de Dados.

Contudo, se o serviço requisitado é sobre mapas, o Provedor de Bromélias faz a requisição para o Provedor de Mapas (Passo M.1). Como é um módulo independente, de

propriedade de uma diferente organização de pesquisa, este faz sua própria verificação de autenticação e autorização (Passo M.2 e M.3) para então enviar a informação requisitada pelo Portal de Bromélias (Passo M.4).

Um processo análogo ocorre se o Provedor de Bromélias necessita informação de outros Provedores (representado pelos Provedores A.1, A.n, passos A1.1 e An.1) para compor a informação requisitada pelo Portal (passos A1.2, A1.3, A1.4).

Do mesmo modo que o Provedor de Bromélias faz requisição da informação para outros Provedores, caso este estiver autenticado e autorizado no Provedor de Mapas, por exemplo, o Provedor de Mapas envia os mapas pedidos através do Provedor de Bromélias tornando o serviço transparente ao usuário.

3.3.1 Projeto do Sistema

3.3.1.1 Base de Dados

Para disponibilizar esses serviços, é necessária uma base de dados e esta foi projetada segundo um Diagrama Entidade Relacionamento baseado no padrão ABCD (ABCD-a, 2005), ou seja, mapeando as informações do padrão descritas em XML, para o modelo relacional. Além disso, foram adicionadas novas entidades com o objetivo de adequar o Diagrama Entidade Relacionamento ao Sistema de Informação de Bromélias (Anexo D), uma vez que nem todas as informações necessárias para catalogação de bromélias estavam presentes no esquema ABCD.

Abaixo, listam-se todas as entidades adicionadas, com suas respectivas funções:

- *Use* tem como principal objetivo, descrever a utilização das bromélias nos campos medicinais, populares, além da área ornamental;
- *EcologicDescription* é responsável por descrever as informações ecológicas de bromélias;
- *MorphologicDescription* tem como intuito principal descrever as informações morfológicas de bromélias, no qual pode-se citar como exemplo o tamanho e o formato das folhas e flores;
- *FlorestaDeCaixeta* contém informações específicas de coleta de bromélias nas Florestas de Caixetas, adequando novas informações ao padrão ABCD (ABCD-a, 2005) e

permitindo aos pesquisadores obter informações mais específicas sobre o local de coleta em comparação ao que propõe o padrão ABCD.

3.3.1.2 Provedor

A Arquitetura do Provedor disponibiliza os seguintes componentes de serviços:

- **Identificadores de Espécie:** Serviço que tem como objetivo inserir, alterar, remover e consultar os dados de todos os identificadores das espécies disponíveis;
- **Nomes Científicos:** Serviço que tem como objetivo inserir, alterar, remover e consultar o nome científico de todas as espécies disponíveis;
- **Dados de Identificação de Espécie:** Serviço que tem como objetivo inserir, alterar, remover e consultar os dados de identificação de uma espécie, como por exemplo o nome científico, primeira epífita, autor;
- **Dados Ecológicos e Morfológicos de Espécie:** Serviço que tem como objetivo inserir, alterar, remover e consultar as informações ecológicas e morfológicas de uma determinada espécie;
- **Identificadores da Imagem:** Serviço que tem como objetivo inserir, alterar, remover e consultar os identificadores das imagens relacionadas de uma determinada espécie;
- **Dados de Coleta:** Serviço que tem como objetivo inserir, alterar, remover e consultar as informações de coleta de uma espécie;
- **Identificadores dos Locais de Coleta:** Serviço que tem como objetivo inserir, alterar, remover e consultar os identificadores do local de coleta relacionado com uma espécie de bromélia;
- **Propriedades da Imagem:** Serviço que tem como objetivo inserir, alterar, remover e consultar todas as propriedades da imagem relacionada a uma espécie de bromélia;
- **Propriedades da Base de Dados:** Serviço que tem como objetivo inserir, alterar, remover e consultar todas as propriedades da base de dados de bromélias.

- Consulta de mapas: serviços que visa através de coordenadas, consultarem a área do mapa, no qual é indicado por um ponto vermelho.

3.3.1.3 Portal

O Portal possui classes em que são manipulados os dados provenientes do Provedor via Serviços. Além disso, nas classes existem métodos que fazem a requisição e recebimento (via *Web Services* e protocolo SOAP ao Provedor) dos dados pedidos pelo usuário. Logo, o Portal faz desde a requisição e recebimento dos dados como também todo o processamento dos dados, para que as informações mostradas nas interfaces sejam ao mesmo tempo completas e de fácil entendimento.

A classe *ShowSpecieInfo* faz a manipulação das informações ecológicas e morfológicas de bromélias, além do processamento dos dados destas e a manipulação dos dados de identificação das bromélias. E na classe *ShowGatherinInfo* contém os métodos para manipulação das informações de coleta das espécies de bromélias.

Além destas, existe a classe *ShowImageInfo* que contém os métodos para manipular as informações das figuras e fotos de bromélias.

A troca de informações entre Portal e Provedor é realizada sempre em formato XML, facilitando o entendimento das informações provenientes do Provedor via Serviços.

3.4 sistema de informação – Arquitetura Baseada em Componentes

Após a definição dos serviços Sistema através da SOA, o próximo passo é definir a arquitetura baseada em Componentes. Isso significa agregar todos os Serviços em componentes. A figura 10 representa esta arquitetura.

O Serviço que gerencia a comunicação entre os Provedores e os portais faz parte agora do Componente Coordenador.

Os Serviços de autenticação e autorização estão agrupados no componente chamado de Segurança.

Os Serviços relacionados com informações de Mapas estão agrupados no componente Provedor de Mapas. O mesmo ocorre com os componentes A0 data e o A1 data que gerenciam aplicações de bromélias.

E por último, temos o componente Portal que é responsável pela interface.

Logo, essa arquitetura permite visualizar todos os serviços em um único componente.

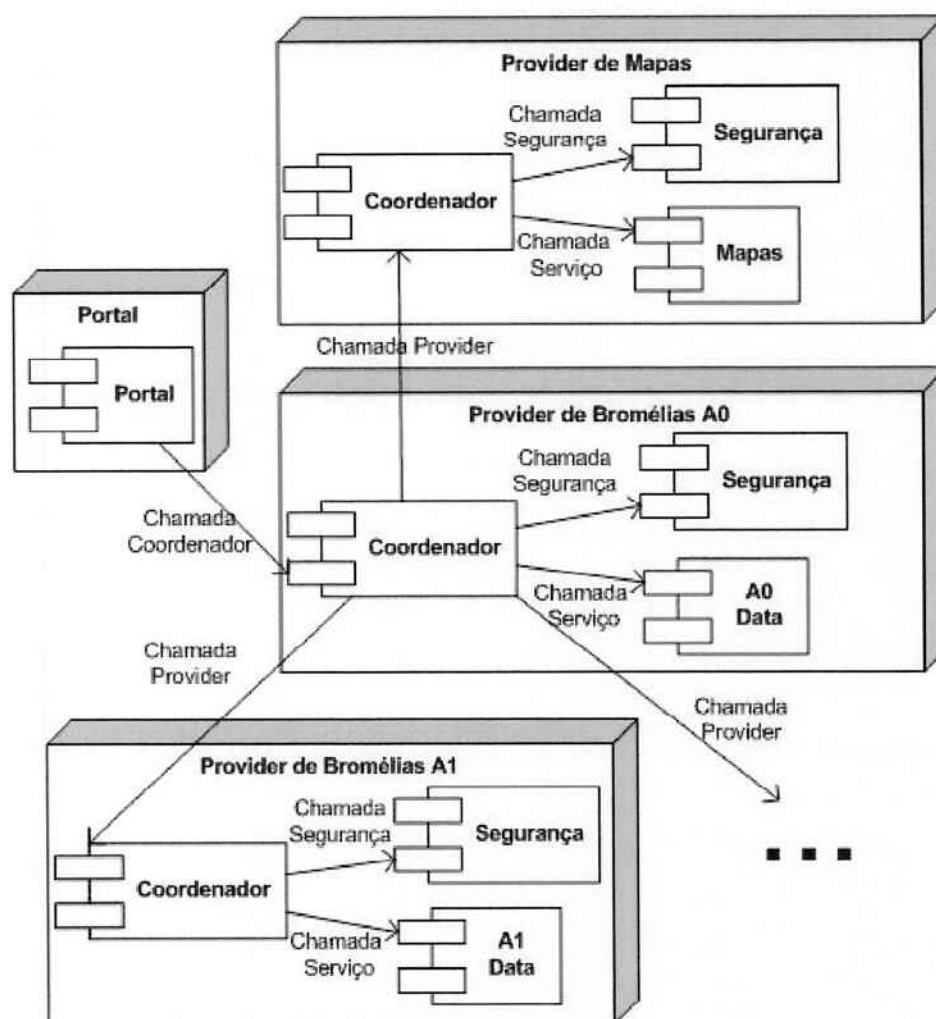


Figura 10: Arquitetura Baseada em Componentes

3.5 sistema de Informação – Tecnologias utilizadas

De posse da arquitetura SOA e de Componentes, implementou-se a seguinte arquitetura tecnológica, utilizando tecnologias existentes (Figura 11). Nota-se que o sistema tem três blocos principais:

- Portal;
- Operação de Segurança, que envolve o coordenador e operadores de autenticação e autorização;
- Operação de Serviços, que compreende o operador de serviço.

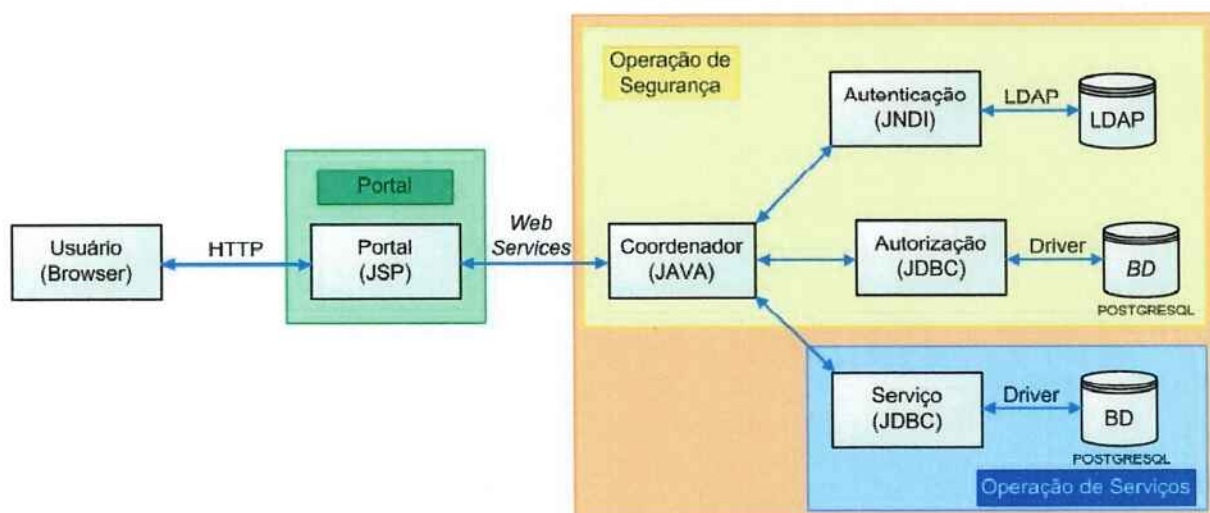


Figura 11: Implementação da arquitetura.

A relação entre estes três blocos é descrita a seguir:

Entre o portal e operador de serviços, existe agora o coordenador (parte da operação de segurança), cujo papel é intermediar todas as operações entre portal e operador de serviços, cumprindo o papel de coordenação de segurança. Assim, quando o usuário acessa o portal, o coordenador se encarrega primeiramente de autenticar o usuário, identificando qual o perfil deste para que, num segundo instante, quando este solicitar algum serviço, o coordenador verifica se o perfil pertencente a este permite tal serviço e caso não esteja autorizado, responder ao usuário negando o acesso ou enviando ao usuário a resposta do serviço pedido.

As informações de autenticação do usuário como *login*, senha e local, se encontram armazenadas em estruturas de diretórios, utilizando o protocolo LDAP para fazer as consultas destes dados. De posse do *login*, senha e local é possível recuperar o perfil do usuário de tal forma a permitir a autorização mais tarde de um serviço. Para isso, utiliza-se o JNDI para comunicar com o protocolo LDAP.

Assim, quando o usuário se autenticou e recuperou seu perfil no sistema, ao se fazer a requisição de um serviço, o coordenador utiliza o perfil do usuário para permitir a autorização

do mesmo, bastando para isso, consultar a tabela de autorização, que descreve os serviços permitidos de um determinado perfil. Esta tabela faz parte de um banco de dados relacional.

Apenas para citar, havia a possibilidade de outras duas arquiteturas parecidas, porém com características diferentes em topologia, mas com funções análogas. A diferença na topologia está em determinar quantos subsistemas o sistema contém.

Ou seja: deixar três subsistemas: portal – operação de segurança – operação de serviços;

Ou dois subsistemas: portal e operação de segurança – operação de serviços;

Ou também: portal – operação de segurança e operação de serviços.

Foram levantados os prós e contras de cada uma das topologias e chegou-se a seguinte tabela comparativa:

Tabela 2: Tabela Comparativa das Topologias

Topologia	Pros	Contras
Portal - Op. Segurança - Op. Serviço	<ul style="list-style-type: none"> • Não modifica o Op. Serviço. Op. Serviço mais flexível através da modificação dos serviços • Mais fácil de implementar • Mais flexível a mudanças • Centraliza os usuários do sistema • Coordenador pode, num projeto posterior, chamar outros Op. Serviços 	<ul style="list-style-type: none"> • Tempo de processamento • Modifica o Portal • Necessidade de implementar transporte de dados mais seguros para os serviços • No caso da queda de um dos três, o sistema fica indisponível
Portal e Op. Segurança - Op. Serviço	<ul style="list-style-type: none"> • Implementa um serviço, que é o pedido da transação, com o retorno dos dados da querie. • Centraliza os usuários do sistema • Coordenador pode, num projeto posterior, chamar outros Provedores 	<ul style="list-style-type: none"> • O módulo Portal e Op. Segurança fica "pesado" e o Op. Serviço, como possui o Banco de Dados, por si só deve ser robusto para suportar as transações • O módulo citado pode ser gerenciado por outra entidade que não criou o provedor, e assim, este pode perder o gerenciamento da base de dados
Portal - Op. Segurança e Op. Serviço	<ul style="list-style-type: none"> • Implementa dois serviços entre Portal - Op. Segurança & Op. Serviço, a saber: autenticação e autorização, sendo que na autorização, já temos o retorno dos dados pedidos caso a autorização esteja ok; 	<ul style="list-style-type: none"> • Muda o provedor, abandonando os serviços já feitos, já que a comunicação entre Op. Segurança e Op. Serviço é feito via java • Cada Op. Serviço tem que ter seu próprio Op. Segurança e seus usuários

Assim, verificamos que na primeira topologia, havia uma grande flexibilidade, porém corria-se risco maior de indisponibilidade, por ter que passar por três máquinas remotas separadas. Na segunda topologia, o empecilho maior é possuir duas máquinas robustas, pois o Op. Segurança e o Op. Serviço precisam de memória e espaço para suas funções, enquanto que a última topologia, que foi escolhida, basta-se ter uma máquina de porte menor e outra de maior capacidade, além de permitir que o responsável da base de dados possa cuidar também do acesso a sua própria base de dados, o que é mais coerente.

4. IMPLEMENTAÇÃO

O Sistema de Informação de Bromélias contempla a implementação três módulos (Figura 9): um portal de bromélias e dois Provedores (um com serviços de informações sobre bromélias e um outro com serviços de informações de mapas).

De acordo com as ferramentas discutidas nos tópicos anteriores, os módulos foram implementados utilizando o diagrama de classes (Anexo A) e seqüências (Anexo C) da UML (RUMBAUGH, 2004).

Primeiramente, foram definidas as transações mapeadas através de serviços do provedor de Bromélias em três categorias relativas aos dados de espécie, de imagem e de coleta. Segue abaixo as transações identificadas.

- Espécie: cadastro, consulta, alteração, exclusão;
- Imagem: cadastro, consulta, alteração, exclusão;
- Coleta: cadastro, consulta, alteração, exclusão.

Existem também outras transações, porém relacionadas à manipulação da segurança. Elas permitem o cadastro, consulta, alteração e exclusão de perfis de acesso.

No caso do provedor de mapas, foi definida apenas uma transação: a de consulta de mapas.

Finalmente, o portal de bromélias faz a interface com todas as transações do provedor de bromélias, enquanto este deve ser capaz de se comunicar com o provedor de mapas.

De posse desses dados, os diagramas de classes foram elaborados (Anexo A) Para facilitar o entendimento, foi separado em três diagramas relativos aos módulos: portal de bromélias (Anexo A, Figura 12), provedor de bromélias (Anexo A, Figura 13) e provedor de mapas (Anexo A, Figura 14).

No diagrama de classes do portal de bromélias (Anexo A, Figura 12), as classes e as interfaces *ProfileSoapBindingStub*, *ProfileServiceLocator*, *ProfileService*, *Profile* foram geradas pela API AXIS (a partir do WSDL gerado) facilitando a comunicação via *Web Services*. Portanto, a classe *ShowProfile* utiliza as interfaces *ProfileService* e *Profile* para instanciar um serviço relacionada a transações de manipulação de perfis de acesso.

De modo análogo, as classes e interfaces *CoordinatorSoapBindingStub*, *CoordinatorServiceLocator*, *CoordinatorService* e *Coordinator* (Anexo A, Figura 12) também foram geradas pela API AXIS (a partir do WSDL) para acessar os serviços de manipulação de dados de espécie, de imagem e de coleta.

Assim, as classes que são responsáveis pela interface com o usuário (classes *ShowImageInfo*, *ShowSpecieInfo*, *ShowGatheringInfo*, *ShowMapInfo*), através da classe *UserAuthorization*, instanciam serviço utilizando as interfaces *CoordinatorService* e *Coordinator*.

É importante ressaltar que a presença das classes *UserAuthorization* e *UserAuthentication* são responsáveis por autorizar e autenticar respectivamente o usuário para que ele possa acessar os serviços do provedor de bromélias.

Passando para o diagrama de classes do provedor de bromélias (Anexo A, Figura 13), temos a classe *Coordinator*, responsável por receber os pedidos de serviços enviados pelas classes *UserAuthorization* e *UserAuthentication* do portal de bromélias e realizar os seguintes passos:

Se for um pedido de autenticação, a classe *Coordinator* verifica, através da classe *LdapAuthentication* se os dados enviados são válidos (esta classe é responsável por validar junto ao LDAP estas informações).

Caso contrário, se for um pedido de autorização, a classe *Coordinator* utiliza a classe *PostGresAuthorization* para que verifique nas tabelas de autorização se os dados de perfil e transação estão corretos. Se não estiver em, a classe *Coordinator* ainda verifica se a transação pedida é para um provedor externo (no nosso caso, o provedor de mapas). Se mesmo assim isso não for correto, a classe *Coordinator* retorna uma mensagem de erro.

Uma vez que a autorização for válida, *Coordinator* verifica com a classe *ProviderManager* qual a transação a ser realizada. Isto significa acessar a classe que tenha a transação desejada (desse modo, serviços relativos a dados de espécies, de imagens ou de coleta são realizadas pelas classes *SpecieInfo*, *ImageInfo* e *GatheringInfo*, respectivamente).

Uma vez a transação foi definida, a classe responsável realiza uma *query* no banco de dados utilizando a classe *Postgres*. Assim, os dados retornados são enviados de volta a classe *Coordinator* para que se finalize o serviço.

Já no caso do diagrama de classes do provedor de mapas (Anexo A, Figura 14), este consta de duas classes: *GetMap* (que é responsável por receber o pedido de serviço do

provedor de bromélias e executar *query* na classe *Postgres*) e *Postgres* (que realiza a *query* no banco de dados).

Utilizando o diagrama de componentes (Anexo B), podemos ver com mais clareza como é o relacionamento dos três módulos (portal de bromélias, provedor de bromélias e provedor de mapas).

Os diagramas de seqüência mostram as interações de forma mais detalhadas. A seguir, breve explicação sobre elas:

No Anexo C, Figura 16 temos a seqüência de autenticação no Provedor de Bromélias. A classe *Coordinator* recebe um pedido de autenticação e utiliza a classe *LdapAuthentication* que verifica o login, local e senha do usuário. De acordo com o retorno de *LdapAuthentication*, *Coordinator* envia mensagem de erro ou sucesso.

Uma vez autenticado, para executar a transação pedida, *Coordinator* deve fazer a autorização e utiliza *PostGresAuthorization* para validar o local e o perfil. Só depois de válido, *Coordinator* executa a transação pelo *ProviderManager* e retorna os dados da transação pedida. Caso contrário, *Coordinator* envia mensagem de erro. Isso é mostrado através do Anexo C, figura 17.

Se no caso a transação pedida for a consulta de mapas (não existe a transação no provedor de bromélias mas em um provedor externo), o *Coordinator* deve chamar o serviço através das classes *Service* e *Client* da API AXIS, como mostra o Anexo C, Figura 18. Se o serviço falhar, a seqüência é mostrada no Anexo C, Figura 19.

Se for uma transação do provedor de bromélias, depois de autorizado, o *Coordinator* encaminha os dados para *ProviderManager* que é o responsável por instanciar a classe correta para a transação pedida. No Anexo C, Figura 20, a classe instanciada é a *SpecieInfo*, mas poderia ser *GatheringInfo* ou *ImageInfo* dependendo da transação desejada. Instanciada a classe desejada, esta acessa *Postgres* para executar a *query* pertinente a transação. O resultado então é repassado até o *Coordinator* para que este finalize o serviço.

Uma vez entendido o funcionamento das seqüências no provedor de bromélias, veremos como é a seqüência de eventos no Portal de Bromélias. Usando o Anexo C, Figura 21, podemos verificar que a interface que já fez a autenticação com o provedor de bromélias através de *UserAuthentication*, utiliza os valores de perfil e local para definir estes valores em *ShowMapInfo*, além de outros parâmetros necessários para a transação. De posse destes

dados, *ShowMapInfo* utiliza *UserAuthorization* para pedir autorização do serviço. *UserAuthorization* chama então o *Coordinator*, idem ao Anexo C, Figura 18.

No caso de transações referentes aos dados de espécies, de imagens e de coleta, a seqüência é análoga, porém utilizam-se classes da API AXIS para acessar os serviços, como mostra o Anexo C, Figura 22.

Finalmente, no Anexo C, Figura 23 pode-se ver a seqüência de eventos quando o provedor de mapas é acessado via serviços. *GetMap* utiliza *Postgres* para exportar as imagens de satélite e as envia para quem fez o pedido de serviço.

Definidos esses diagramas citados, partiu-se para a implementação. Definiu-se o modelo das interfaces de *login*, consulta de bromélias e de imagens de satélites (Anexo E). Configurou-se o LDAP e o Postgresql.

O Diagrama Entidade Relacionamento (Anexo D) baseado no padrão ABCD foi utilizado para a montagem de *scripts* de criação tabelas da base de dados de Bromélias.

5. DISCUSSÃO DOS RESULTADOS

5.1 Resultados obtidos

A Arquitetura proposta foi validada, na qual foi implementada uma primeira versão do Sistema de Informação para Compartilhamento de Bases de Dados Distribuídas de Biodiversidade, composto de um Portal, um Provedor de Informações de Bromélias, um Provedor de Mapas de Satélites, cada qual com seu sistema de segurança, responsável pela autorização e autenticação. Também foi gerada uma Base de Dados, sendo que esta última foi projetada de acordo com o padrão internacional de catalogação de espécies ABCD. Todas as informações necessárias sobre bromélias, para testes do Sistema, foram obtidas com a pesquisadora Mariana Carvalhaes (ESALQ, 2005), que contribuiu também para a definição dos Serviços necessários para o Sistema.

Além disso, baseado neste projeto foram produzidos diversos artigos, dentre os quais se podem citar:

- Artigo (NAGAMINE et al., 2004) e pôster que foi apresentado no 12º SICUSP (Simpósio Internacional de Iniciação Científica da Universidade de São Paulo);
- Artigo (CORRÊA et al., 2004) e pôster para o I2TS (*Internacional Information and Telecommunication Technologies Symposium*) 2004;
- Artigo (SUZUKI et al., 2005) e apresentação oral no EFITA/WCCA 2005, realizado na cidade de Vila Real, Portugal.

5.2 Sugestão de Continuidade do Projeto

Em relação ao Sistema de Informação para Compartilhamento de Bases de Dados Distribuídas de Biodiversidade, este pode não só contribuir para a informatização da catalogação de espécie, como também para permitir uma maior troca de informações entre os agentes envolvidos, eliminando assim informações redundantes ou desnecessárias.

Dessa forma, através da Arquitetura Orientada a Serviços, toda a pesquisa sobre espécies de bromélias será facilitada, uma vez que o Portal se encarregará de fazer a pesquisa

em todos os Provedores disponíveis, obtendo assim uma pesquisa abrangente e rápida através dos Serviços disponíveis. Tudo isso, poupará esforços dos pesquisadores e demais interessados, uma vez que este não terá que realizar pesquisas em todas as bases de dados bromélias de forma manual, deixando isso para o Sistema de Informação.

Sob o ponto de vista de Engenharia, deve-se ressaltar a aplicação de técnicas de Engenharia de Software na especificação do Sistema, baseados em UML e Diagrama Entidade Relacionamento. Além disso, a utilização de ferramentas de desenvolvimento *open-source* facilitou a implementação nos diferentes sistemas além de ter custo reduzido para sua implantação. Outro aspecto importante foi a modelagem do sistema utilizando o diagrama de classes e de sequência, o que permitiu ter uma visão geral de todas as funcionalidades do sistema, além de permitir uma implementação mais rápida e segura, tendo em vista toda preparação realizada anteriormente.

Uma continuidade dessa pesquisa pode surgir baseada no DER (Anexo D) do padrão ABCD, criando a possibilidade da utilização desta para futuros Sistemas de catalogação de outras espécies na área de biodiversidade, contribuindo para a troca de informações entre os diversos agentes envolvidos. Além disso, pode surgir uma outra linha de pesquisa baseada no gerenciamento da informação sobre biodiversidade nos Portais, na qual os pesquisadores poderão manipular as informações, ou seja, realizar cruzamentos de informações, permitindo assim a criação de relatórios, com informações mais detalhadas e de maior abrangência sobre a espécie pesquisada.

O projeto levantou, também, especificações de segurança dentro de uma arquitetura de banco de dados distribuídas de Biodiversidade. Para isso, foi fundamental a definição da arquitetura de segurança proposta, bem como a implementação desta utilizando o Sistema de Informação para Compartilhamento de Bases de Dados Distribuídas de Biodiversidade. Para esta definição, o estudo da arquitetura DiGIR foi valiosa, pois permitiu verificar e readequar este sistema para suportar o controle de acesso.

Além disso, o projeto também abordou o estudo, a especificação e a implementação da transferência de imagens e arquivos via *Web Services*, sendo esse muito importante, uma vez que a arquitetura orientada a serviço deve contemplar a transferência deste tipo de informação. Logo, através da divulgação desse tipo de informação, as pesquisas podem se tornar mais abrangentes e completas, pois os futuros usuários do sistema poderão trocar desde imagens de espécies, como obter mapas de satélites. Futuramente, até outros tipos de mídias.

Este projeto é um passo dado em direção a um sistema que permitirá que colecionadores, produtores, pesquisadores e formuladores de políticas de conservação planejem projetos futuros referentes à redução do impacto ambiental e proteção da biodiversidade, além de centralizar estas informações contemplando, as necessidades específicas de cada grupo de usuários e pesquisadores do sistema (de acordo com seus perfis de acesso), além de controlar o acesso a essas informações tão importantes.

6. BIBLIOGRAFIA

ABCD-a. *Access to Biological Collection Data . CODATA Working Group on Biological Collection Data Access*. Informações sobre ABCD. Disponível em: <<http://www.bgbm.org/TDWG/CODATA/Schema/>>. Acessado em 03 de março de 2005.

ABCD-b. CIRCA – *Communication & Information Resource Centre Administrator*. Informações sobre ABCD. Disponível em: <http://forum.belspo.be/Public/irc/bccm/gbif_wfcc_workshop/library?xt=1>. Acessado em 03 de março de 2005.

APACHE. *Apache Jakarta Tomcat Project*. Apresentação do projeto Jakarta Apache Tomcat. Disponível em: <<http://jakarta.apache.org/tomcat/>>. Acessado em 03 de março de 2005.

AVH. *Council of Heads of Australian Herbaria - Austrália*. Apresenta um exemplo de uma base de dados distribuída com dados sobre a botânica da Austrália. Disponível em: <<http://www.chah.gov.au/avh.html>>, Acessado em 03 de março de 2005.

AXIS. *Apache Jakarta Tomcat Axis*. Apresentação do projeto Jakarta Apache Axis, implementação do protocolo SOAP (*Simple Object Access Protocol*). Disponível em <<http://ws.apache.org/axis>>. Acessado em 03 de março de 2005.

BERKELEY. *University of California - Berkeley*. Apresenta o Projeto do Sistema de Informações em Rede sobre Mamíferos. Disponível em <<http://dlp.cs.berkeley.edu/manis/index.html>>. Acessado em 03 de março de 2005.

COFAN-NUNES, J.V. 2002. Bromélias. IN: Simões, L.L & Lino, C.F.(org.). *Sustentável Mata Atlântica - A exploração de seus recursos florestais*. 213p. Editora SENAC São Paulo.

CORRÊA, Pedro Luiz Pizzigatti; SUZUKI, Vitor Hugo; NAGAMINE, Takae Terasé; GUIMARÃES, Rodrigo Gioia; CARVALHAES, Mariana. *A Service Oriented Information System to Manage Bromélia Distributed Database*. In: EFITA WCCA 2005, Vila Real, Portugal, 2005. Vila Real : *Book of Abstracts*, 2005. p. 57.

CVS - *Concurrent Version System*. Apresentação da Comunidade CVS e o respectivo projeto. Disponível em <<https://www.cvshome.org/>>. Acessado em 30 de agosto de 2004.

DARWINCORE. *Species Analyst – Darwin Core V2*. Informações sobre o padrão DarwinCore. Disponível em: <<http://tsadev.speciesanalyst.net/documentation/ow.asp?DarwinCoreV2>>. Acessado em 03 de março de 2005.

DIGIR. *DiGIR Open Source Project*. Apresentação do projeto DiGIR. Disponível em: <<http://digir.sourceforge.net/>>, Acessado em 03 de março de 2005.

ECLIPSE. *Eclipse Consortium*. Apresentação do Ambiente de Desenvolvimento Integrado (IDE) Eclipse. Disponível em: < <http://www.eclipse.org/>>. Acessado em 03 de março de 2005.

EMBRAPA Embrapa Monitoramento por Satélite. Disponível em: < <http://www.cnpm.embrapa.br/> >. Acessado em 03 de março de 2005.

ESALQ. ESALQ - Escola Superior de Agricultura "Luiz de Queiroz". Disponível em: < <http://www.esalq.usp.br/>>. Acessado em 03 de março de 2005.

FAPESP. Sistema de Informações Ambiental Biota. Apresenta informações sobre o Sistema de Informações Biota, desenvolvido para cadastramento de informações levantadas por pesquisadores do projeto Biota da FAPESP. Disponível em< <http://www.sinbiota.org.br/>>. Acessado em 03 de março de 2005.

HTTP. *World Wide Web Consortium (W3C) – HTTP (Hypertext Transfer Protocol)*. Documentação sobre HTTP. Disponível em <<http://www.w3.org/Protocols/>>. Acessado em 30 de Agosto de 2004.

IBM. UDDI. *Universal Discovery, Description, and Integration*; Disponível em: <<http://www.ibm.com/service/uddi/standards.html>>. Acessado em 03 de março de 2005.

JAVA. *Java Technology* Apresentação da plataforma Java. Disponível em: <<http://java.sun.com/>>. Acessado em 03 de março de 2005.

JAVA 2 PLATFORM ENTERPRISE EDITION (J2EE). *Java 2 Platform, Enterprise Edition (J2EE) Overview*. Apresentação da plataforma J2EE. Disponível em : <<http://java.sun.com/j2ee/overview.html>>. Acessado em 03 de março de 2005.

JAVASCRIPTS. Netscape. Documentação do JavaScript. Disponível em <<http://developer.netscape.com/docs/manuals/javascript.html>>. Acessado em 30 Dde agosto de 2004.

JNDI. SUN, *SUN Microsystems, Inc.* API para conexão do LDAP. Disponível em <<http://java.sun.com/developer/technicalArticles/Programming/jndi/index.html>>. Acessado em 03 de março de 2005.

JSP. *Java Server Pages Technology*. Apresentação da tecnologia JSP. Disponível em: <<http://java.sun.com/products/jsp/>>. Acessado em 03 de março de 2005.

LDAP. IETF - *The Internet Engineering Task Force. Reference for Comments* . Documentação sobre o LDAP. Disponível em: <<http://www.ietf.org/rfc/rfc2251.txt>>. Acessado em 03 de março de 2005.

LINUX. *The Linux Home Page*. Apresentação do Sistema Operacional Linux. Disponível em: <<http://www.linux.org>>. Acessado em 03 de março de 2005.

MAPSERVER. *MapServer Homepage*. Apresentação do MapServer. Disponível em: <<http://mapserver.gis.umn.edu/>>. Acessado em 06 de março de 2005.

NAGAMINE, Takae Teras; GUIMARÃES, Rodrigo Gioia; FERREIRA, Marcelo Succi de Jesus; CARVALHAES, Mariana; SARAIVA, Antonio Mauro; CORRÊA, Pedro Luiz Pizzigatti. Sistema de segurança para compartilhamento de bases de dados distribuídas de biodiversidade. In: SIMPÓSIO INTERNACIONAL DE INICIAÇÃO CIENTÍFICA DA UNIVERSIDADE DE SÃO PAULO., São Paulo, SP, 2004. SIICUSP 2004: resumos.. São Paulo: USP, 2004. res.1018 on-line.

OPENLDAP. Projeto OpenLDAP. Disponível em <www.openldap.org>. Acessado em 03 de março de 2005.

PHP Comunidade PHP. Disponível em <www.php.net>. Acessado em 30 de agosto de 2004..

POSIX. *IEEE POSIX Certification Authority*. Apresentação do Posix. Disponível em <<http://standards.ieee.org/regauth/posix/>>. Acessado em 03 de março de 2005.

POSTGRESQL. PostgreSQL. Apresentação da base de dados Postgresql. Disponível em:<<http://www.postgresql.org/>> . Acessado em 03 de março de 2005.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. *The Unified Modeling Language Reference Manual*. Estados Unidos: Addison Wesley, 2003. 550p.

SOA. – *Service-Oriented Architecture*. Definição da Arquitetura orientada a serviços. Disponível em <http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html>. Acessado em 30 de agosto de 2004.

SOS Mata Atlântica. Parabólicas On-line. Revista on-line sobre questões sócios-ambientais. Disponível em: <<http://www.socioambiental.org/website/parabolicas/edicoes/edicao40/reportag/pg10.htm>>. Acessado em 03 de março de 2005.

STAL, M. *Web Services: Beyond Component-Based Computing*. *Communications of ACM*, v. 45 n.10, p.71-76, october, 2002.

SUZUKI, Vitor Hugo; NAGAMINE, Takae Terasa; GUIMARÃES, Rodrigo Gioia; CARVALHAES, Mariana; CORRÊA, Pedro Luiz Pizzigatti. *Bromélia Information System*. In: *INTERNATIONAL INFORMATION AND TELECOMMUNICATION TECHNOLOGIES SYMPOSIUM.*, São Carlos, Brasil, 2004.

TDWG. *Taxonomic Database Working Group*. Apresentação do grupo TDWG. Disponível em: <<http://www.tdwg.org/>>. Acessado em 03 de março de 2005.

TOMCAT. *Apache Jakarta Tomcat Project*. Apresentação do projeto Jakarta Apache Tomcat. Disponível em <<http://jakarta.apache.org/tomcat/>>. Acessado em 30/08/04.

UNIX. *The Unix System*. Apresentação do Sistema Operacional Unix. Disponível em: <<http://www.unix.org/>>. Acessado em 03 de março de 2005.

W3C-a. *World Wide WEB Consortiun*. Informações sobre o padrão *Web Services*. Disponível em: <<http://www.w3.org/2002/ws/>>. Acessado em 03 de março de 2005.

W3C-b. *World Wide WEB Consortiun*. Informações sobre o SOAP. Disponível em: <<http://www.w3.org/TR/SOAP>>. Acessado em 03 de março de 2005.

W3C-c. *World Wide WEB Consortiun*. Informações sobre o WSDL. Disponível em: <<http://www.w3.org/TR/wsdl.html>>. Acessado em 03 de março de 2005.

W3C-d. *Word Wide WEB Consortium*. Informações sobre protocolos. Disponível em: <<http://www.w3.org/>>. Acessado em 03 de março de 2005.

WEBBEE. Departamento de Engenharia de Computação da Escola Politécnica da Universidade de São Paulo. Informações sobre o projeto WebBee - rede de informações sobre biodiversidade brasileira em abelhas nativas. Disponível em: <<http://www.webbee.org.br>>. Acessado em 03 de março de 2005.

WINDOWS. *Microsoft Windows Family Home Page*. Apresentação do Sistema Operacional Windows. Disponível em: <<http://www.microsoft.com/windows>>. Acessado em 03 de março de 2005.

XML. XML.com:XML From the Inside Out. Informações sobre o XML. Disponível em: <<http://www.xml.com>>. Acessado em 03 de março de 2005.

XP. *Extreme Programming. A gentle introduction*. Apresentação da Metodologia *Extreme Programming*. Disponível em: <<http://www.extremeprogramming.org>>. Acessado em 03 de março de 2005.

ANEXO A : Diagrama de Classes

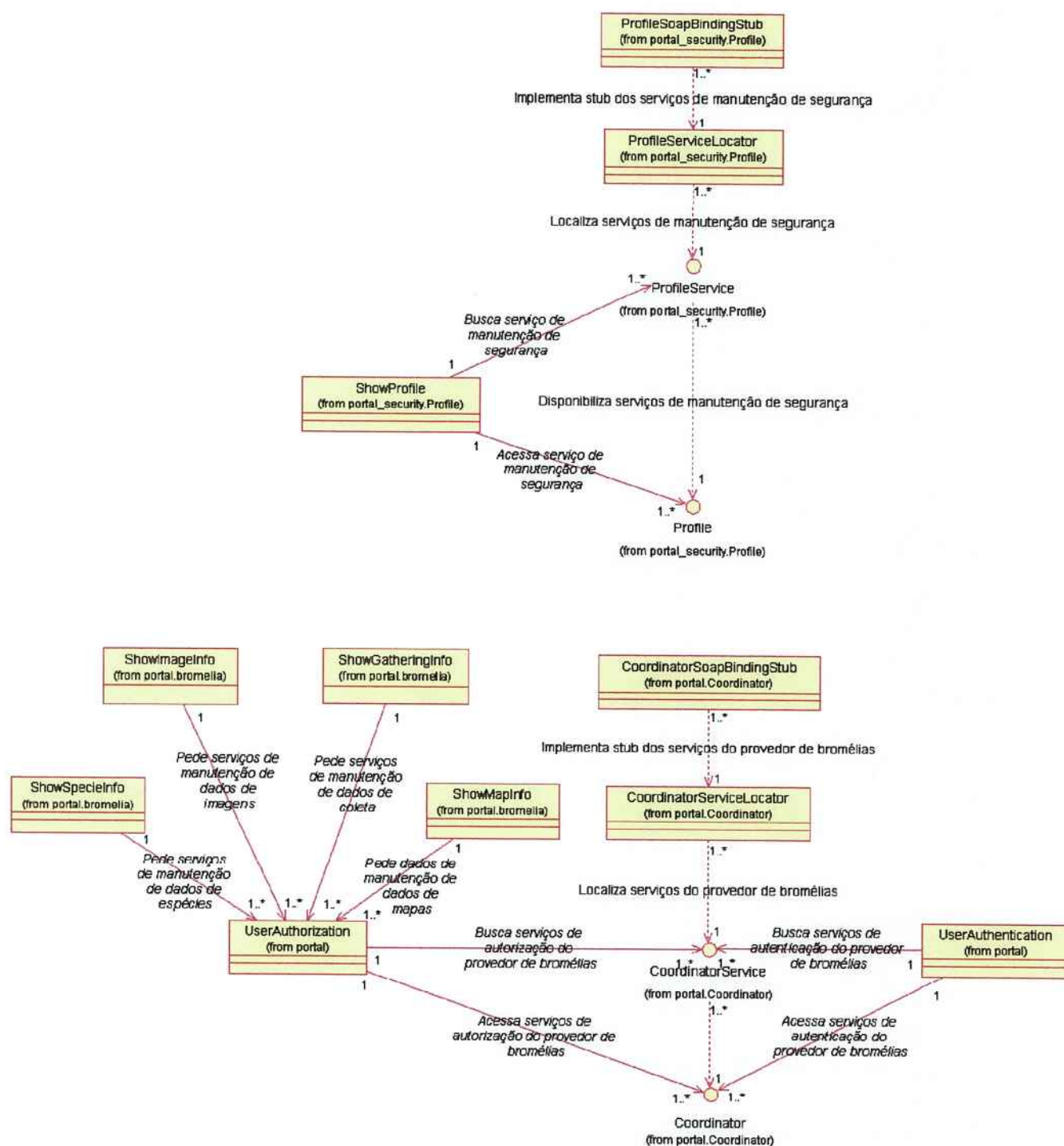


Figura 12: Diagrama de Classes do Portal de Bromélias

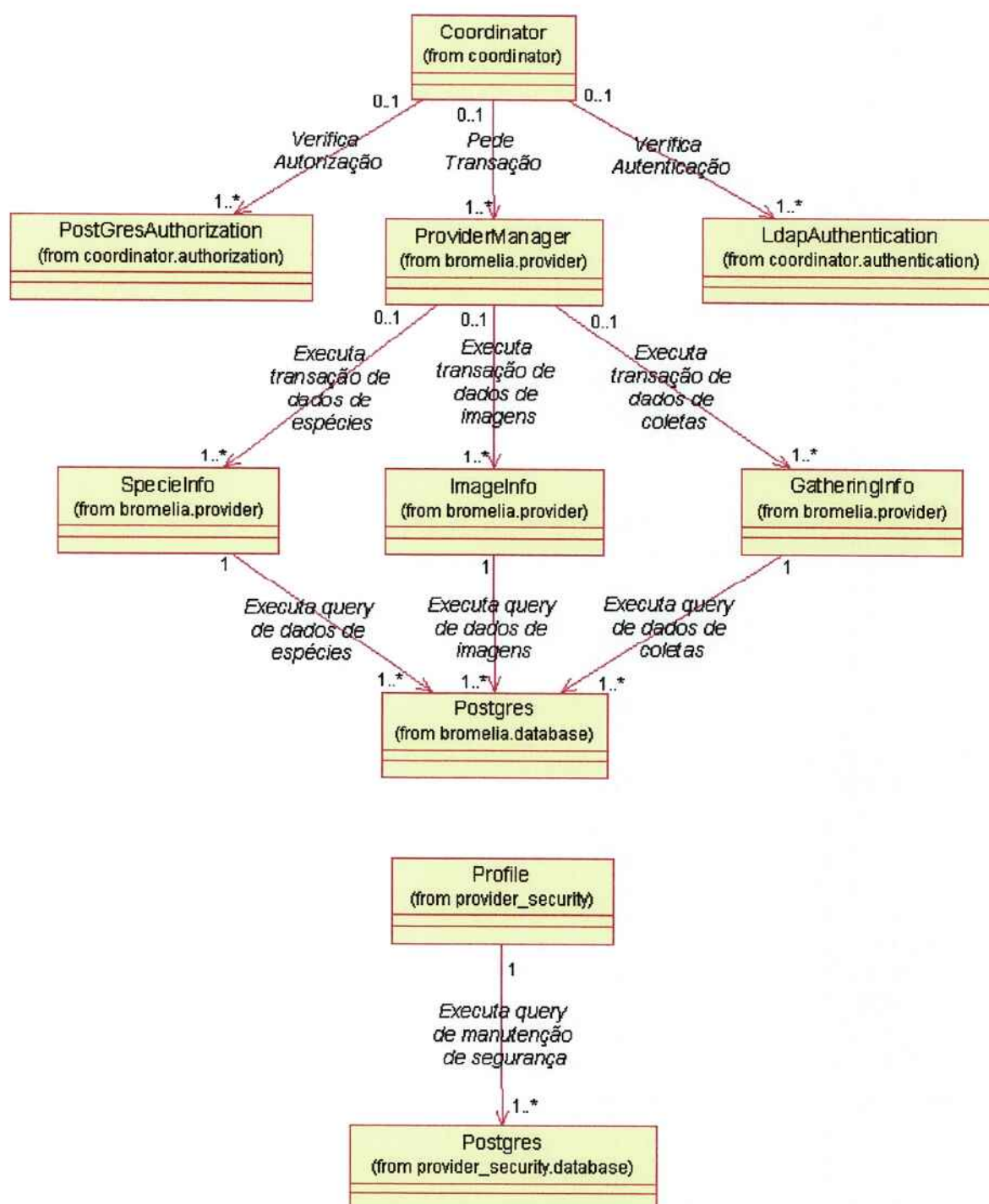


Figura 13: Diagrama de Classes do Provedor de Bromélias



Figura 14: Diagrama de Classes do Provedor de Mapas

ANEXO B : Diagrama de Componentes

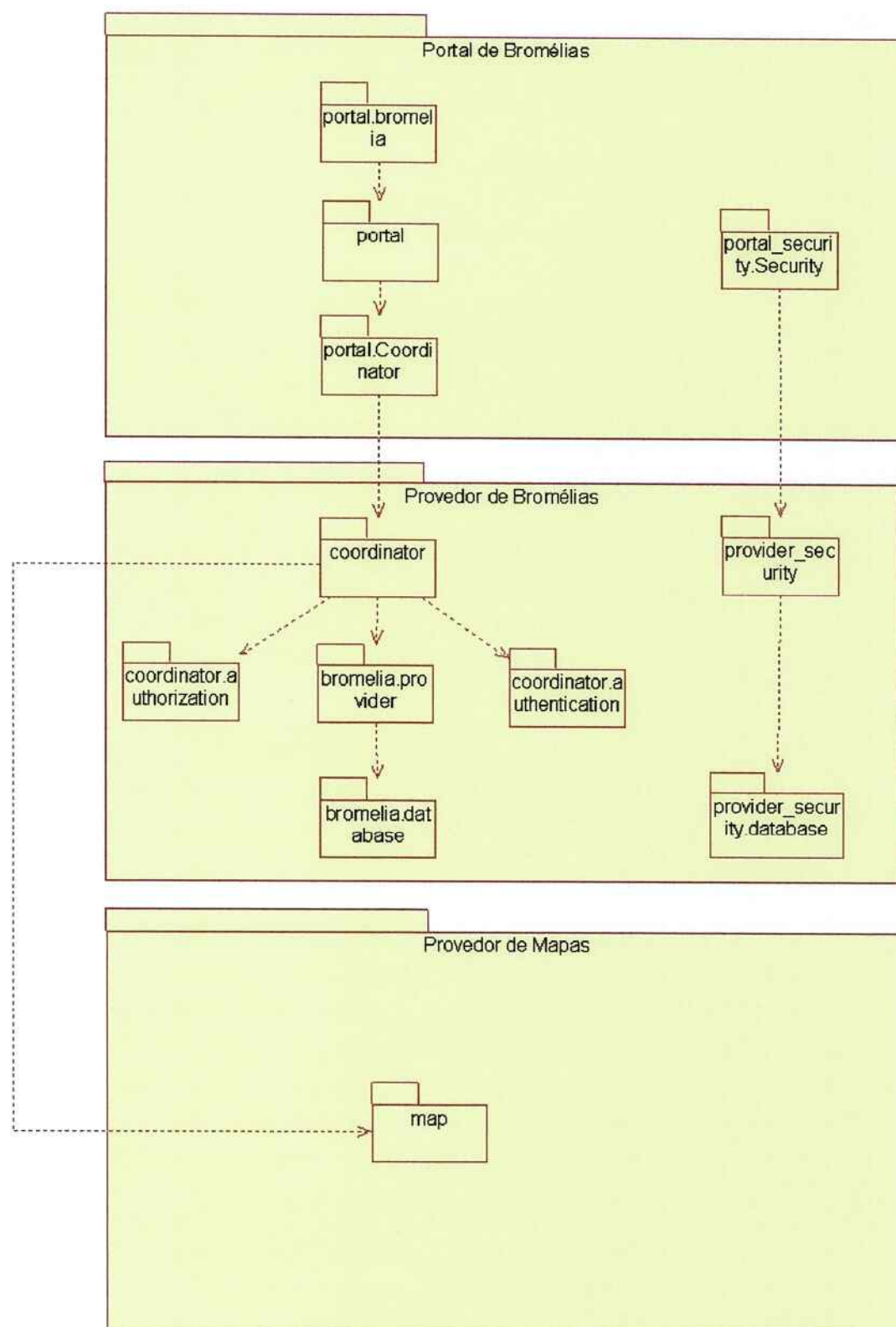
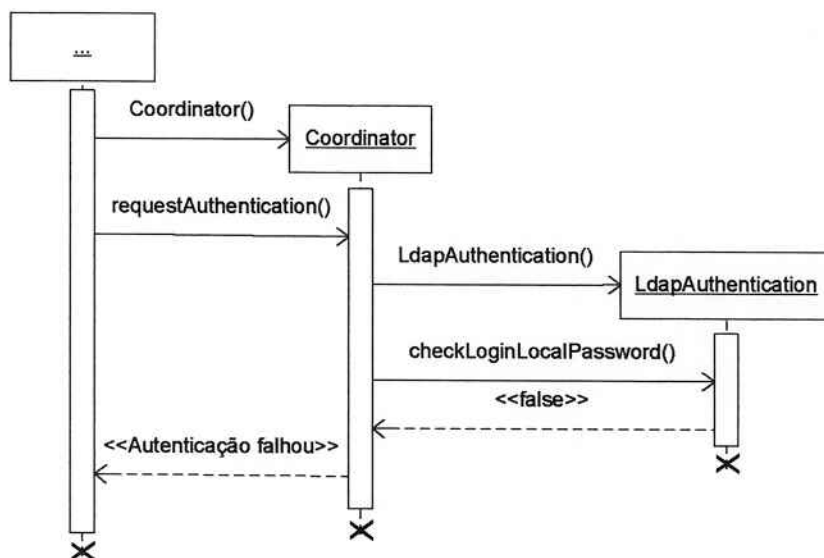


Figura 15: Diagrama de Componentes do Sistema de Bromélias

ANEXO C : Diagrama de Sequência – Provedor de Bromélias

Coordenador recebendo pedido de autenticação inválida



Coordenador recebendo pedido de autenticação válida

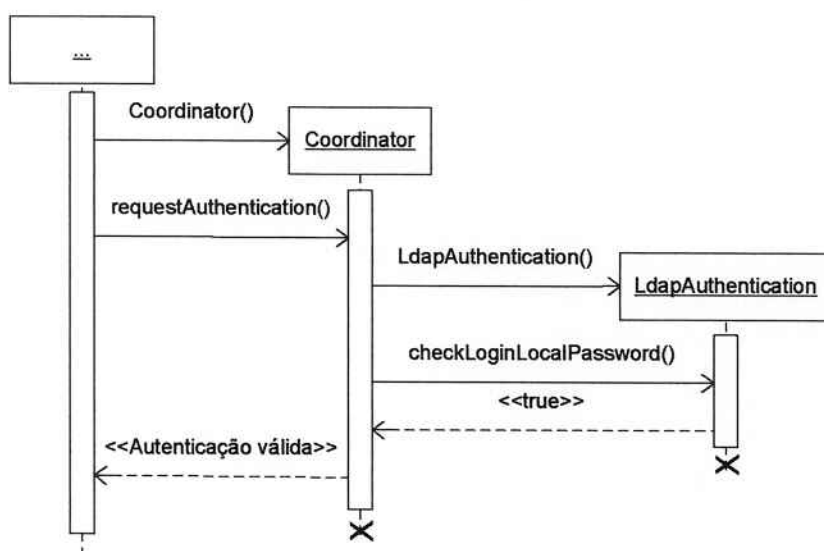
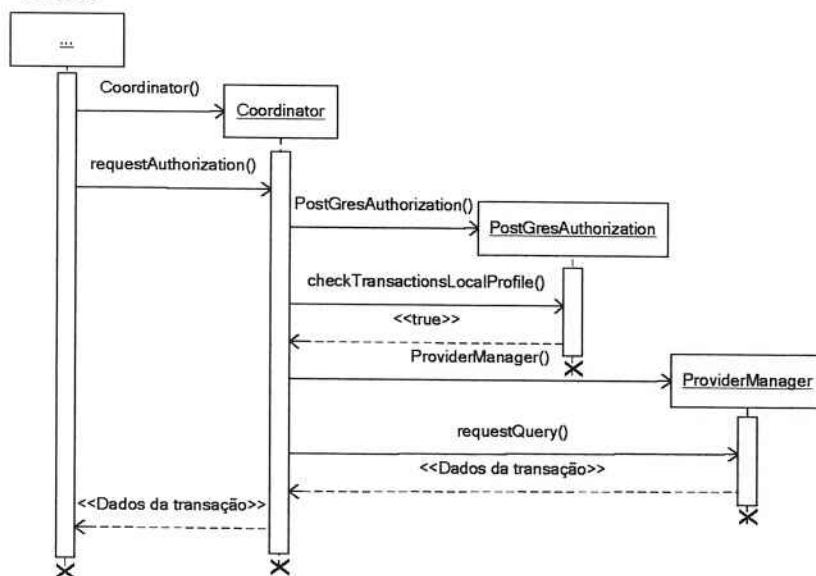


Figura 16: Diagramas de Sequência do Provedor de Bromélias - Classe *Coordinator* recebendo autenticação inválida e válida

Coordenador recebendo pedido de autorização de transação de dados de espécies / imagens / coletas



Coordenador recebendo pedido de autorização inválida.
Transação, local e profile inválidos

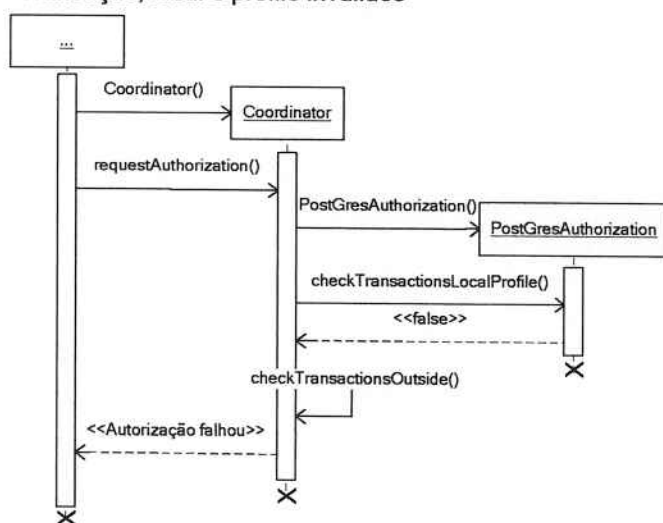


Figura 17: Diagramas de Sequência do Provedor de Bromélias - Classe *Coordinator* recebendo autorização válida e inválida

Coordenador recebendo pedido de autorização para transação de dados de mapas

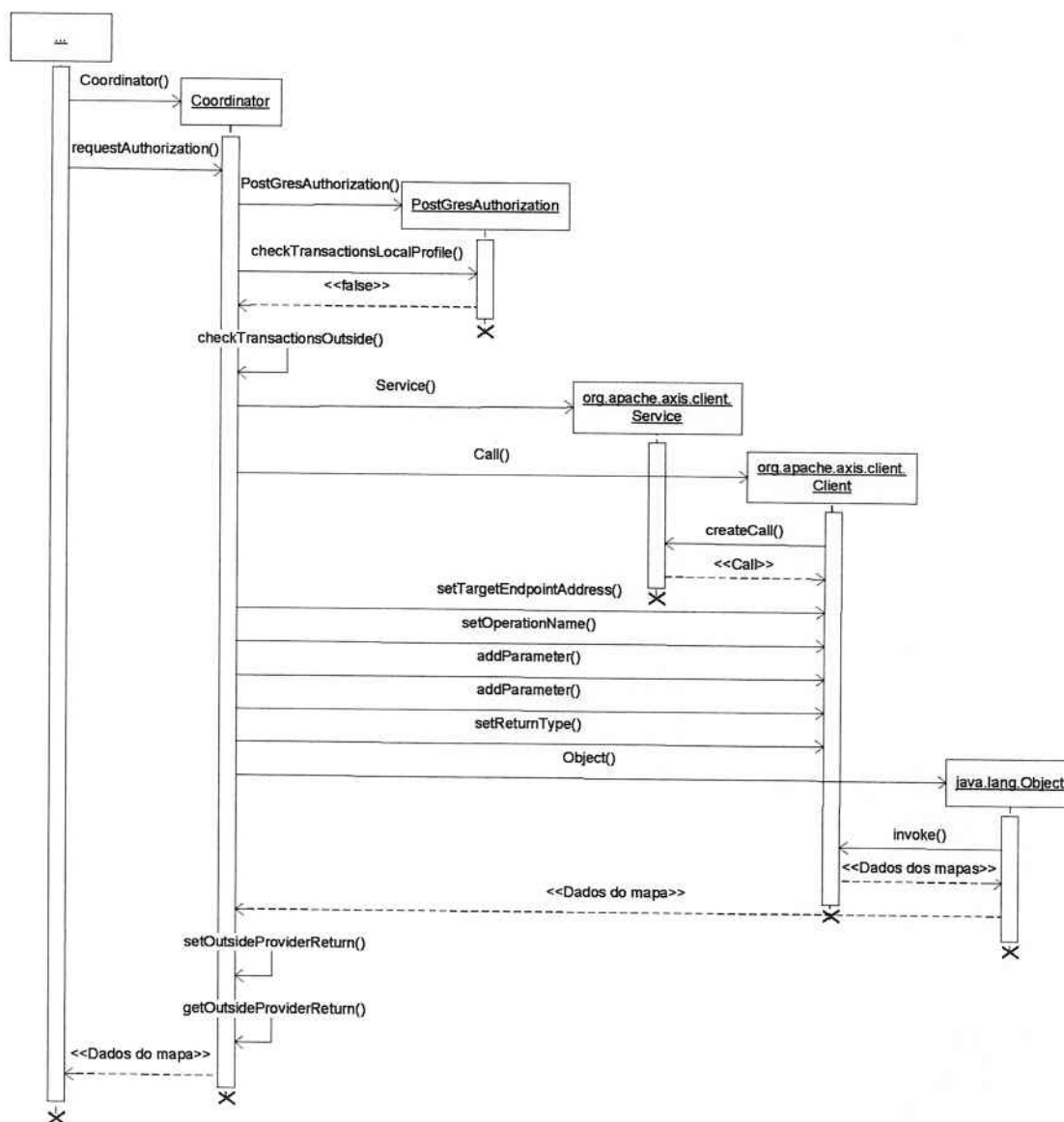


Figura 18: Diagramas de Seqüência do Provedor de Bromélias - Classe *Coordinator* acessando serviço do provedor de mapas

Coordenador recebendo pedido de autorização para transação de dados de mapas.
 Erro ao acessar serviço de mapas.

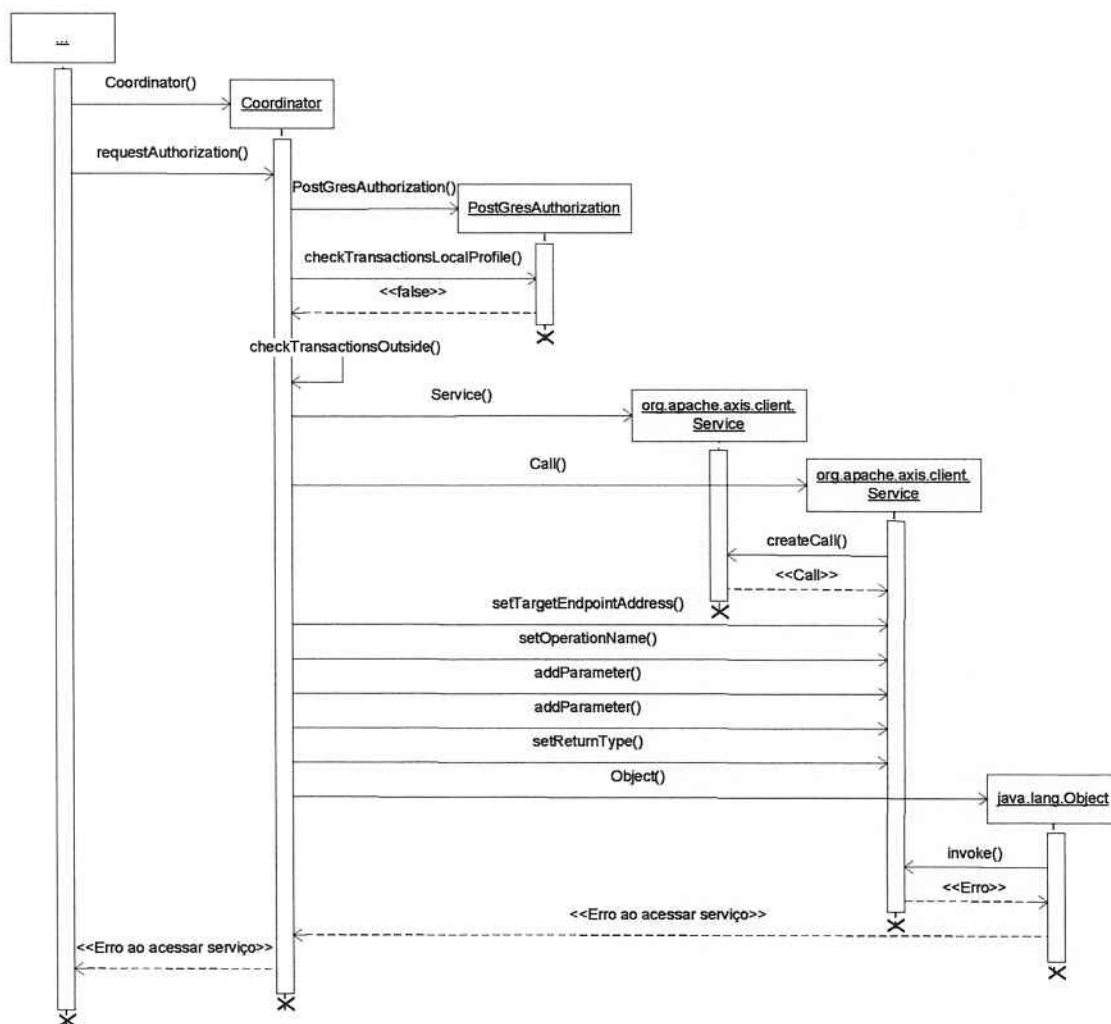


Figura 19: Diagramas de Seqüência do Provedor de Bromélias - Classe *Coordinator* acessando serviço do provedor de mapa. Erro ao acessar serviço

ProviderManager recebendo pedido de transação do Coordinator. Exemplo análogo para GatheringInfo, ImageInfo e outras transações para SpecieInfo (consulta, cadastro, alteração, exclusão)

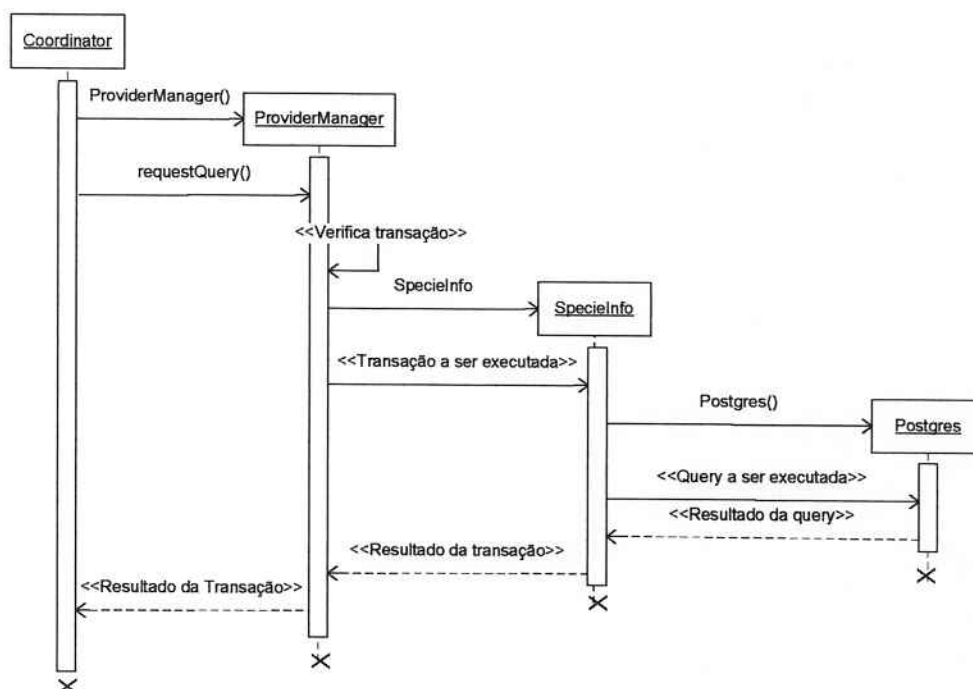


Figura 20: Diagramas de Sequência do Provedor de Bromélias - Classe *Coordinator* executando transação do provedor de Bromélias..

Interface pedindo consulta de mapas

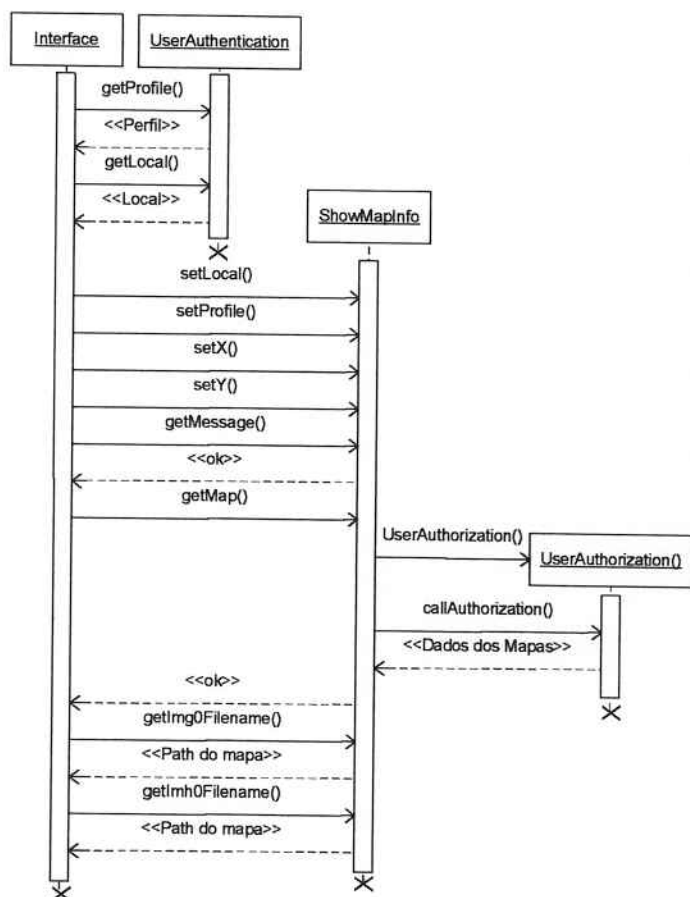


Figura 21: Diagramas de Seqüência do Provedor de Bromélias - *ShowMapInfo* acessando serviço do provedor de mapa.

Acessando serviço do Provedor de Bromélias utilizando o portal.
 Para ShowGatheringInfo, ShowImageInfo, ShowEspecie, a sequência é análoga.

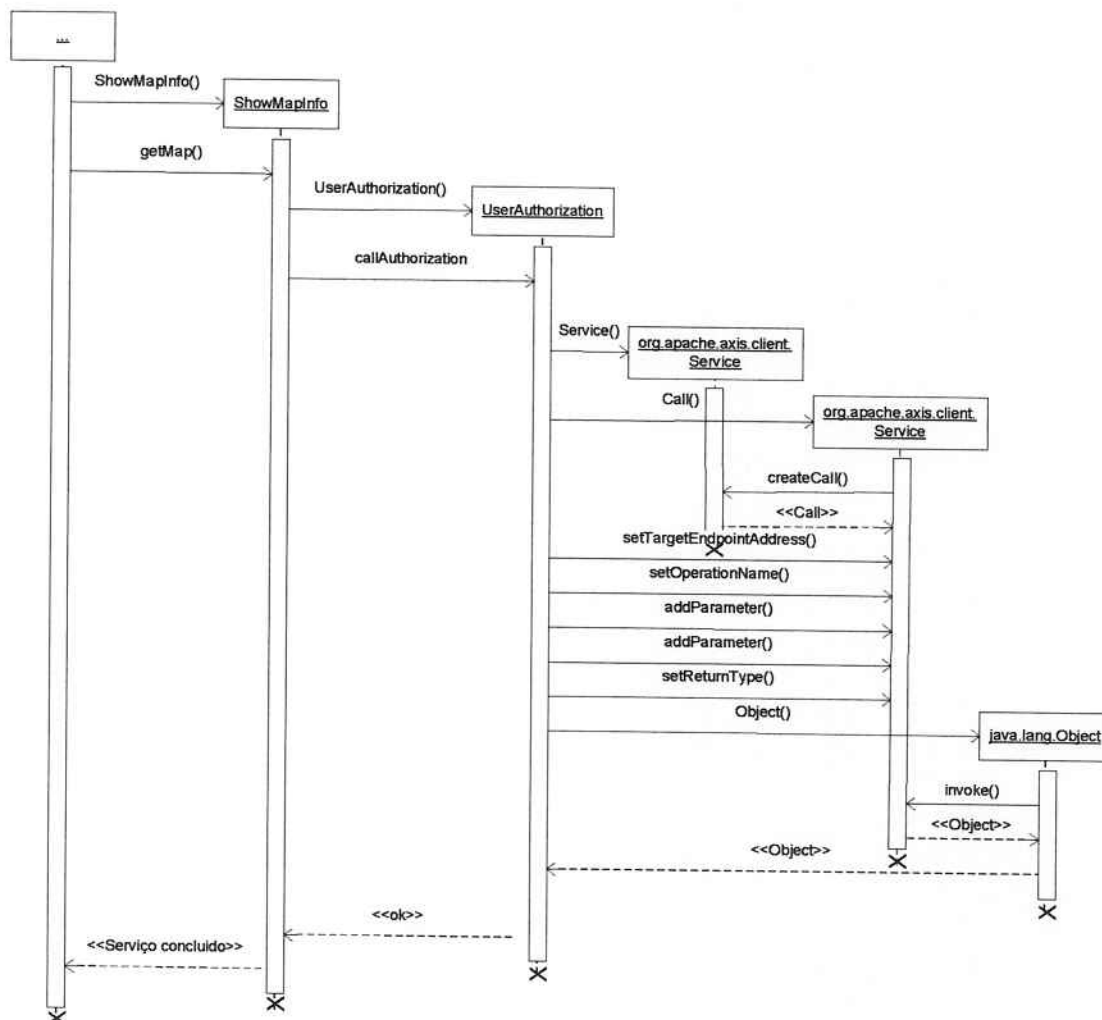


Figura 22: Diagramas de Seqüência do Portal de Bromélias acessando serviço do provedor de Bromélias

GetMap recebendo pedido de mapa

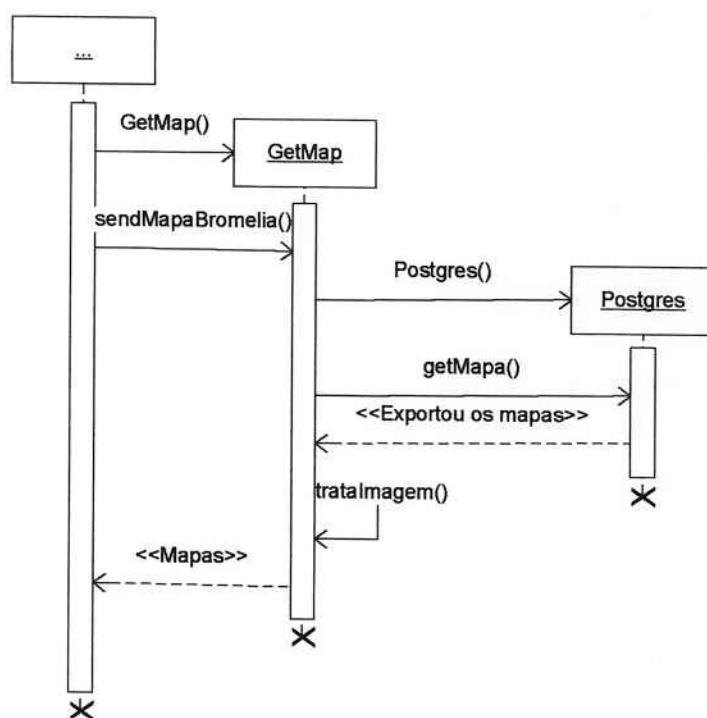


Figura 23: Diagramas de Seqüência do Provedor de Mapas - Classe *GetMap* executando o serviço de envio de mapas.

ANEXO D: Diagrama Entidade Relacionamento

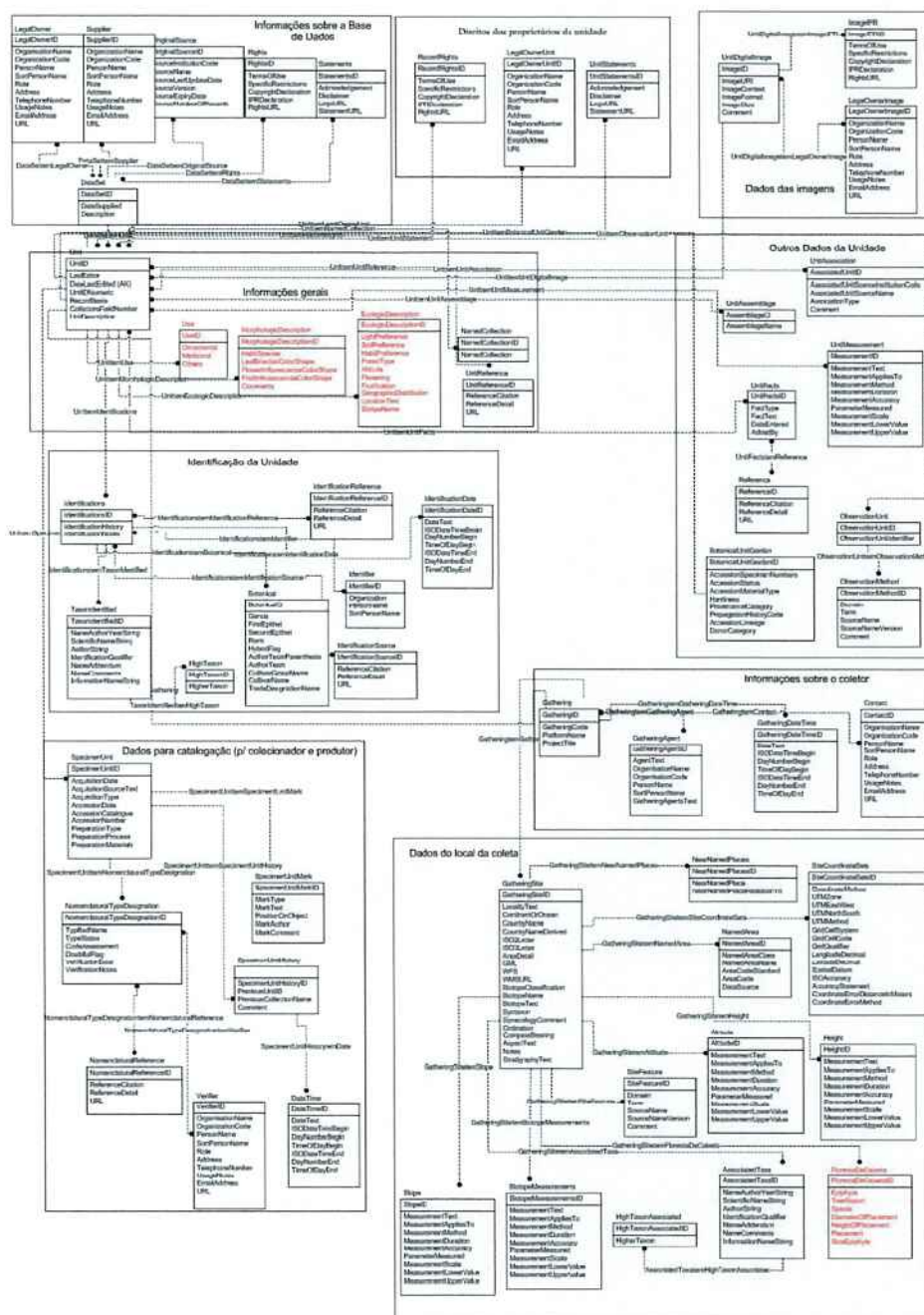


Figura 24: Diagrama Entidade Relacionamento da Base de Dados

ANEXO E: Interface do Sistema



Bem vindo ao Sistema de Informação de Bromélias

Por favor, digite login, local e
senha

Login:

Local:

Senha:

Caso seja visitante, use login:
guest, local: l=usp,c=br, senha:
1234

Figura 25: Interface de *Login*

**Universidade de São Paulo**EPUSP / Departamento de Engenharia de Computação e Sistemas Digitais
ESALQ / Departamento de Engenharia Florestal

Bem vindo ao Sistema de Informação de Bromélias

Selecione sua opção:	Consultar Espécie
	Inserir Espécie
	Alterar Espécie
	Excluir
Imagens da Espécie:	Inserir Imagem
	Consultar Imagens
	Alterar Imagens
	Excluir Imagens
Dados e Locais de Coleta:	Inserir Dados de Coleta
	Inserir Local de Coleta
	Consultar Dados e Locais de Coleta
	Alterar Dados e Locais de Coleta
	Excluir Dados e Locais de Coleta
Controle de segurança:	Manter Segurança

Figura 26: Interface de Consulta

Consultar Mapa do Local da Coleta

Dados do Mapa

Longitude: -47.62

Latitude: -20.20



Figura 27: Interface de Consulta de Imagem de Satélite

APÊNDICE A: Descrição do CD do Projeto de Formatura

O CD do Projeto de formatura é composto pelos seguintes diretórios:

/Manual de Instalação do Sistema

Contém o manual detalhado para a instalação de todo o ambiente de desenvolvimento, bem como a instalação do Sistema de Informação para Compartilhamento de Bases de Dados Distribuídas de Biodiversidade.

/Fontes/Portal

Diretório que contém todas as fontes do Portal de Bromélias

/Fontes/Provedor

Diretório que contém todas as fontes do Provedor de Bromélias

/Fontes/Provedor_Mapas

Diretório que contém todas as fontes do Provedor de Mapas

/Fontes/SQL

Diretório que contém todos os scripts SQL's para gerar a Base de Dados de Bromélias

/Mapas

Diretório que contém todos os mapas de satélites utilizados para o Provedor de Mapas

/Programas e Bibliotecas

Diretório que contém todos os programas, bibliotecas e Sistema Operacional para a Instalação do Ambiente de Desenvolvimento e Ambiente de Produção.

/Monografia

Diretório que contém a monografia do Projeto de Formatura