UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RAFAEL HENKIN

Interface de Consultas Analíticas para Bases de Dados de Biodiversidade

Trabalho de Graduação

Profª Drª Renata de Matos Galante
Orientadora

Profª Drª Carla M. D. S. Freitas
Co-orientadora

Porto Alegre, julho de 2010

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Henkin, Rafael

Interface de Consultas Analíticas para Bases de Dados de Biodiversidade / Rafael Henkin. – Porto Alegre: UFRGS, 2010.

39 f.: il.

Trabalho de Conclusão (graduação) – Universidade Federal do Rio Grande do Sul. Curso de Ciência da Computação, Porto Alegre, BR–RS, 2010. Orientadora: Renata de Matos Galante; Co-orientadora: Carla M. D. S. Freitas.

1. Análise de dados. 2. Sistemas : integração. 3. Biodiversidade. 4. Banco de dados. I. Galante, Renata de Matos. II. Freitas, Carla M. D. S.. III. Interface de Consultas Analíticas para Bases de Dados de Biodiversidade.

“Olá, como vai? Quase sempre vai bem
Era de se esperar.”
— PÚBLICA

AGRADECIMENTOS

Agradeço a meus pais, Hélio e Ida, e meu irmão Marcelo, por todos os dias da semana que convivemos juntos; a meus tios, avós e primos, principalmente pelos finais de semana que convivemos juntos.

Agradeço ao pessoal do Centro de Processamento de Dados da UFRGS, especialmente à Zaida e à Carla, pela oportunidade e responsabilidades dadas há quase 4 anos.

Aos colegas que me acompanharam em algum momento do curso, pela companhia e apoio.

Às professoras Carla e Renata, pela orientação do trabalho, e à Samantha, ao Denison e ao Pedro pelo apoio no trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

SQL	Structured Query Language
SGBD	Sistema de Gerenciamento de Banco de Dados
TDWG	Taxonomic Database Working Group
PHP	PHP: Hypertext Processor
GBIF	Global Biodiversity Information Facility
OLAP	On-line Analytical Processing
ROLAP	Relational On-line Analytical Processing
HOLAP	Hybrid On-line Analytical Processing
XML	Extensible Markup Language
IPC	Inter-process communication
CRAN	Comprehensive R Archive Network
FIFO	First-in First-out

LISTA DE FIGURAS

RESUMO

Este trabalho apresenta uma interface para a integração de análise e armazenamento em um sistema de informação de biodiversidade. Descreve-se o sistema em questão, chamado *TaxonomyBrowser* e desenvolvido para auxiliar a organização de dados coletados por biólogos, além de sistemas de informação de biodiversidade existentes e suas características. Após, descreve-se os principais conceitos de análise de dados do ponto de vista do banco de dados, o ambiente R, utilizado pelos biólogos para fazer análise dos dados, e as principais formas que possibilitam a integração. Ao final, a implementação da interface é detalhada, assim como suas vantagens e desvantagens.

Palavras-chave: Análise de dados, sistemas : integração, biodiversidade, banco de dados.

Using LaTeX to Prepare Documents at II/UFRGS

ABSTRACT

This work presents an interface for the integration of analysis and storage in a biodiversity information system. The system in question, which is called TaxonomyBrowser and was developed to help organizing data collected by biologists, is described, as well as existing biodiversity systems and their characteristics. Afterwards, we describe the main concepts of data analysis in databases, the R environment, used by the biologists for data analysis, and the main methods that can be used for the integration. Finally, we detail the interface implementation, along with its advantages and disadvantages.

1 INTRODUÇÃO

A expansão da *web* impulsionou o desenvolvimento de portais para facilitar o acesso a dados de biodiversidade, assim como o desenvolvimento de sistemas para organizar a coleta destes dados. Entretanto, a maioria destes portais não oferece opções para análise dos dados, geralmente restrita a soluções de consultas analíticas em bancos de dados.

Este trabalho consiste na implementação de uma interface que faça a integração entre um sistema chamado *TaxonomyBrowser* (TAXONOMYBROWSER, 2010), desenvolvido em um projeto na Universidade Federal do Rio Grande do Sul, e o ambiente *R* (http://www.r-project.org), um *software* utilizado para análise estatística. O *TaxonomyBrowser* é um sistema de informação de biodiversidade, projetado para auxiliar o gerenciamento de dados coletados por biólogos. A interface deve fornecer uma forma de realizar uma análise sobre esses dados sem a necessidade de conhecimento da linguagem de programação R, possibilitando também que esta análise seja feita à distância.

Apesar da integração de análise e armazenamento em bases de dados de biodiversidade trazer benefícios aos usuários dos sistemas, os maiores portais de acessos a dados de biodiversidade não oferecem este tipo de serviço, principalmente por não incluírem efetivamente dados que possam ser analisados. São os casos do portal do *Global Biodivesity Information Facility* e do *Species 2000*, entre outros estudados. A ideia do *TaxonomyBrowser* é também armazenar características e medidas diversas de espécimes coletados e não somente informações de coleta, como data de coleta e localização.

Desta forma, a integração fica restrita a *softwares* específicos. A partir do ambiente R, por exemplo, é possível acessar uma base de dados. Entretanto, não há uma ferramenta que permita o acesso completo à linguagem R no PHP. É neste ponto que surge a necessidade da interface, através da qual será possível realizar análise estatística sobre os dados armazenados pelos biólogos no *TaxonomyBrowser*.

Neste documento, o *TaxonomyBrowser* é apresentado no capítulo 2, com a arquitetura do sistema e detalhes do funcionamento das interfaces já implementadas. No capítulo 3, é feito um estudo comparativo entre portais de dados de biodiversidade, considerando diversos aspectos como usabilidade, opções de busca e exportação dos dados para uso em

outras ferramentas. O capítulo 4 descreve a análise de dados do ponto de vista do banco de dados, o ambiente R e as possibilidades para integração do ambiente R com a linguagem PHP. A interface de consultas analíticas é apresentada no capítulo 5, assim como um breve estudo de caso com os dados disponíveis no sistema no momento da escrita do texto. O capítulo 6 apresenta as conclusões sobre o trabalho, além das perspectivas de desenvolvimento da interface após a implementação inicial.

2 DESCRIÇÃO DO SISTEMA TAXONOMYBROWSER

O *TaxonomyBrowser* surgiu a partir da necessidade do desenvolvimento de um sistema de informação de biodiversidade para auxiliar biólogos no gerenciamento de dados que descrevem espécimes coletados (CAÑETE et al., 2010). Como o armazenamento dos dados, por parte dos biólogos, não era consistente, a análise de um conjunto de dados envolvia uma manipulação longa e complexa de várias planilhas de dados. A proposta do *TaxonomyBrowser* é facilitar a coleta dos dados para análise, além de integrar no próprio sistema ferramentas de análise. Assim, neste capítulo são apresentadas a arquitetura do sistema, o modelo de dados e a aplicação.

2.1 Arquitetura do Sistema

O sistema pode ser separado em 4 componentes principais, ilustrados na figura 2.1:

- a interface de gerência da coleção, na qual é possível gerenciar os dados dos espécimes;
- a interface de consulta visual utilizando o *Google Maps/Earth*, que permite consultas georeferenciadas a dados da coleção;
- a interface de consultas analíticas, desenvolvida neste trabalho e
- a base de dados da coleção, armazenada em um SGBD *MySQL*.

O sistema prevê o acesso de usuários com diferentes autorizações: um usuário comum não tem acesso à interface de gerência de dados, enquanto um biólogo pode cadastrar novas espécimes e também consultá-las utilizando as interfaces disponíveis. A escolha pelo SGBD *MySQL* foi feita devido ao grande conjunto de ferramentas que facilitam o desenvolvimento e administração de sistemas, a licença *GNU General Public License* e sua portabilidade para diferentes sistemas operacionais (CAÑETE et al., 2010).

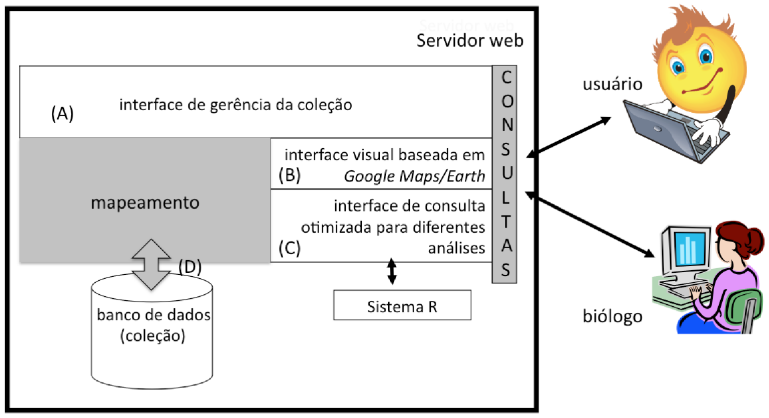

Figura 2.1: Arquitetura do sistema *TaxonomyBrowser*

2.2 Modelo do banco de dados

O banco de dados do sistema *TaxonomyBrowser* foi projetado para armazenar dados de coleta e características de espécimes. A estrutura da base incorpora elementos do padrão Darwin Core (TDWG, 2009), que favorecem a interoperabilidade entre sistemas que armazenam dados de biodiversidade. O modelo parte da ideia de um nodo taxonômico, que é a representação de um nível na classificação taxonômica de seres vivos (CAÑETE et al., 2010). Estes nodos são organizados hierarquicamente, formando uma árvore taxonômica, mostrada na figura 2.2.

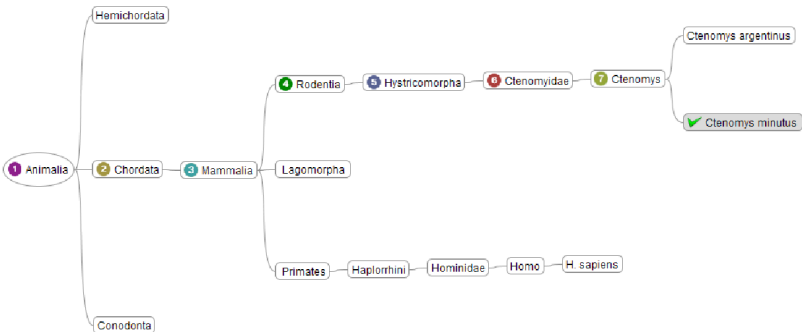

Figura 2.2: Exemplo de árvore taxonômica

No modelo Entidade-Relacionamento da base de dados, cada entidade *Nodo Taxonômico* contém o nome do nível no qual ele está situado (Reino, Filo, Classe, Ordem, Subordem, Família, Gênero ou Espécie), o *valor* deste nodo (o nome da espécie, no caso do nível ser *Espécie*) e uma referência a um nodo taxonômico imediatamente superior a ele na árvore. Uma entidade *Espécime* tem como atributos dados de coleta e tem uma relação com um nodo do nível *Espécie* Além disso, cada *Espécime* tem uma ou mais *Características* morfométricas, taxonômicas e filogenéticas e cada uma das características está associada a uma determinada entidade *Edição*, que contém a informação da unidade de

medida utilizada. Além de valores normais, as características podem armazenar imagens e sons. A última entidade importante diretamente relacionada ao conteúdo do sistema é a *Bibliografia*, que está relacionada a um determinado nível taxonômico e contém informações básicas como título e autor da obra. Para simplificar a visualização, a figura 2.3 mostra o modelo E-R sem os atributos.

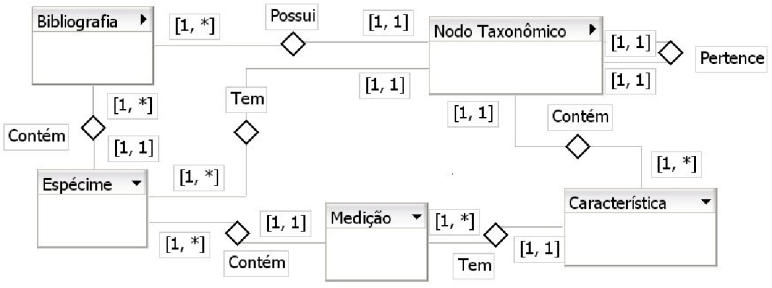

Figura 2.3: Arquitetura do sistema *TaxonomyBrowser*

2.3 Aplicação

A aplicação do *TaxonomyBrowser* foi implementada sobre uma arquitetura de camadas Model-View-Controller (Modelo-Visualização-Controlador), na qual os controladores fazem a ligação dos dados (o modelo) com a interface (a visualização). Cada controlador implementado representa uma entidade do modelo. O usuário que acessar a interface de gerência de dados irá interagir com uma visualização que, por sua vez, é suportada por uma classe controladora. Esta classe controladora é a responsável pela alteração dos dados na base.

Além da divisão das entidades entre classes controladoras, a interface também foi separada em 4 componentes na implementação: a interface de gerência, a interface de consulta, a interface sobre o mapa e a interface de consultas analíticas. As interfaces de consulta e sobre o mapa são o ponto de partida de um usuário que deseja utilizar a interface de consultas analíticas.

A interface de consultas oferece um editor de consultas com operadores básicos de comparação: igual (==), maior (>), menor (<), diferente (!=), maior ou igual (>=), menor ou igual (<=), similaridade (*like*) e intervalo (*between*). O usuário pode escolher as características cadastradas no sistema e também dados de coleta. O resultado pode ser exibido como tabela para a análise ou sobre um mapa, quando as coordenadas geográficas da coleta estiverem disponíveis.

A interface sobre o mapa, ilustrada na figura 2.4, que também serve para que a consulta inicial seja refinada, foi implementada utilizando a API do *Google Maps*, que através da linguagem *JavaScript* permite a consulta a bancos de dados e desenho sobre um mapa.

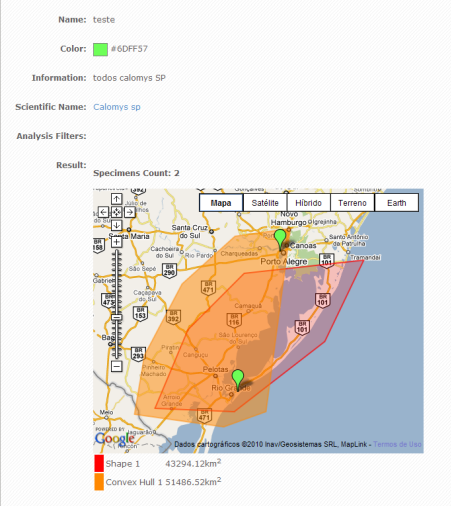

Figura 2.4: Arquitetura do sistema *TaxonomyBrowser*

O usuário também tem à sua disposição opções para personalização do resultado. Ele pode, por exemplo, escolher as cores da marcação nos mapas. Entre as ferramentas para consulta sobre o mapa, existe a possibilidade de demarcar áreas com polígonos, agrupando espécimes. Também é possível exibir a envoltória convexa de uma consulta, que é a demarcação da área ocupada pelas espécimes que estão no resultado da consulta.

2.4 Considerações finais

A proposta inicial do *TaxonomyBrowser* era resolver o problema do armazenamento dos dados e também facilitar a consulta a eles para que sejam importados em outros sistemas utilizados pelos biólogos. Para este fim, foram implementadas as interfaces de gerência e de consulta visual. Com a base de dados e o sistema pronto, foi possível idealizar novas extensões para auxiliar no trabalho dos biólogos.

A interface de consultas analíticas, proposta e desenvolvida neste trabalho, é uma destas extensões. Utilizando a estrutura existente do *TaxonomyBrowser*, a interface foi desenvolvida sobre PHP e preparada para suportar qualquer ferramenta de análise de dados. No entanto, a integração com o *TaxonomyBrowser* foi feita para utilizar o sistema R, que oferece um ambiente e linguagem de programação para análise estatística e é o sistema utilizado pelos biólogos participantes do projeto.

3 REVISÃO BIBLIOGRÁFICA

A disseminação e a coleta de informação apresentam diversos desafios, tanto para quem desenvolve sistemas de coleta como para instituições que desejam compartilhar seus dados. Sistemas que agregam dados e oferecem consulta a estes dados têm preocupação com o desempenho do sistema, enquanto as instituições geralmente precisam exportar os dados de acordo com algum padrão. Neste contexto, temos sistemas especializados em informações sobre biodiversidade, que possuem soluções para alguns problemas específicos da área, além de organizações que estabelecem padrões para troca destas informações. Por exemplo, o sítio do *Biodiversity Information Standards* [1] (em português: Padrões de Informação de Biodiversidade), grupo que desenvolve padrões e protocolos para dados de biodiversidade, lista centenas de projetos relacionados a dados de biodiversidade. Os principais sistemas e portais, estudados para o desenvolvimento deste trabalho, são apresentados e comparados a seguir.

3.1 Padrões para troca de informação

Existem diversas organizações e iniciativas para estabelecer padrões para troca de informação de dados de biodiversidade, de forma que estes dados sejam compatíveis com as mais variadas tecnologias existentes hoje em dia. Um destes grupos é o já mencionado *Biodiversity Information Standards* (também autodenominado TDWG), que foi formado com o objetivo de estabelecer uma colaboração internacional entre projetos de bancos de dados biológicos (TDWG, 2009).

Entre as suas atividades, o TDWG mantém diversos grupos de trabalho com o intuito de promover a discussão para o desenvolvimento dos padrões requisitados pela comunidade. A participação de especialistas de todas as áreas envolvidas - biologia, tecnologia da informação, entre outros - é inclusive encorajada pelo grupo.

O principal padrão estabelecido pelo TDWG é o *Darwin Core*, uma extensão do

[1] http://www.tdwg.org/

Dublin Core, padrão ISO para metadados de objetos digitais como vídeos, imagens e textos. Os metadados são os dados que descrevem outros dados: palavras-chave, resumos e informações de arquivo, entre outros. O padrão *Darwin Core* foi concebido para descrever e facilitar o compartilhamento dados de biodiversidade a partir de um conjunto de regras de formatação destes dados. Estas regras definem tanto um significado único para termos com múltiplos usos, como também os valores possíveis para alguns atributos. Entre os sistemas apresentados a seguir, existem alguns que utilizam o *Darwin Core* para troca de dados ou até mesmo como forma de exportar os resultados.

Além disso, o TDWG também definiu uma classificação sobre projetos de biodiversidade, diferenciando os portais que agregam dados dos portais que fornecem os dados. Esta classificação inclui:

- portais agregadores de dados, que coletam dados de diversas fontes;
- portais de índices, que fornecem listas de provedores de dados;
- portais provedores, que compartilham dados de pesquisas ou coleções próprias;
- padronizadores, que contribuem para a definição de novos padrões para dados;
- e portais facilitadores, que de alguma forma facilitam o acesso a dados de biodiversidade, não necessariamente via *web*.

Todos os sistemas estudados estão listados no site do TDWG e classificados de acordo com os termos acima.

3.2 Sistemas e Portais de Dados de Biodiversidade

3.2.1 Species 2000

O Species 2000 é um programa que reúne diversas instituições do mundo e possui como objetivo a criação de um catálogo de todas as espécies de animais, plantas, fungos e micróbios do mundo. Os dados são fornecidos pelas instituições participantes e coletados pelo sistema do projeto. Cada instituição executa um conjunto de softwares denominado *SPICE* fornecido pelo próprio Species 2000, com instruções de uso em (SPICE, 2006), que permite que o servidor central do Species 2000 colete os dados dos participantes. Desta forma, mesmo que a origem dos dados seja diferente, os dados coletados estão no formato correto.

A natureza do Species 2000 é diferente do projeto deste trabalho, pois o primeiro, de acordo com a classificação do TDWG, é um portal agregador de dados, enquanto o

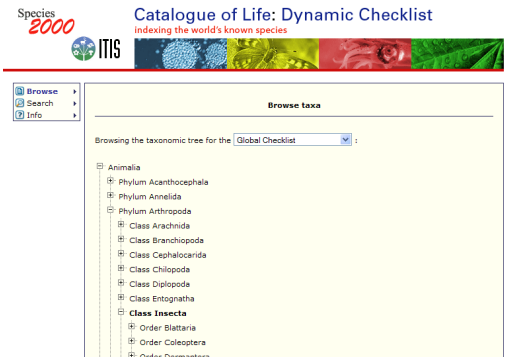

Figura 3.1: Navegação na árvore taxonômica no portal do Species 2000

projeto deste trabalho se aproxima mais de um provedor de dados. Entretanto, existem algumas similaridades, como o fato de os dois sistemas apresentarem ao usuário a literatura relacionada a cada espécie registrada na base e a possibilidade de navegação na árvore taxonômica. As consultas por texto no Species 2000 são feitas sobre os nomes das espécies, incluindo nome científico e nome comum, e também sobre a árvore taxonômica. Uma busca por *Canis* retorna o gênero *Canis*, além de espécies que contêm a palavra *Canis* no nome, como a planta *Aloe canis* e o chacal *Canis adustus*.

Além disso, a descrição do desenvolvimento do *SPICE* em (JONES et al, 2000) apresenta algumas soluções para melhorar a eficiência do sistema, que é um fator importante para qualquer projeto, seja de grande ou pequeno porte, como é este trabalho. Como o Species 2000 coleta dados de diversas bases, uma destas melhorias é restringir a quantidade de bases acessadas, diminuindo assim o tempo que o usuário espera pelos resultados de uma consulta.

3.2.2 GBIF - Global Biodiversity Information Facility

O GBIF, assim como o Species 2000, é um portal[2] agregador de dados. É uma iniciativa multinacional, cujo objetivo, descrito em (GBIF, 2009), é oferecer gratuitamente e abertamente dados de biodiversidade. Assim como no Species 2000, os participantes devem fornecer os dados nos formatos esperados pelo sistema do GBIF, o *Darwin Core* ou o ABCD (*Access to Biological Collection Data*), um padrão para compartilhamento de coleções completas de dados biológicos. Um software chamado *Integrated Publishing Toolkit* (IPT) foi desenvolvido com o objetivo de facilitar a publicação de dados pelos participantes e é oferecido gratuitamente às instituições registradas.

Ao contrário do Species 2000, que é somente um catálogo de espécies, os dados contidos no portal de dados do GBIF incluem registro de espécimes. Entretanto, como o

[2]http://data.gbif.org

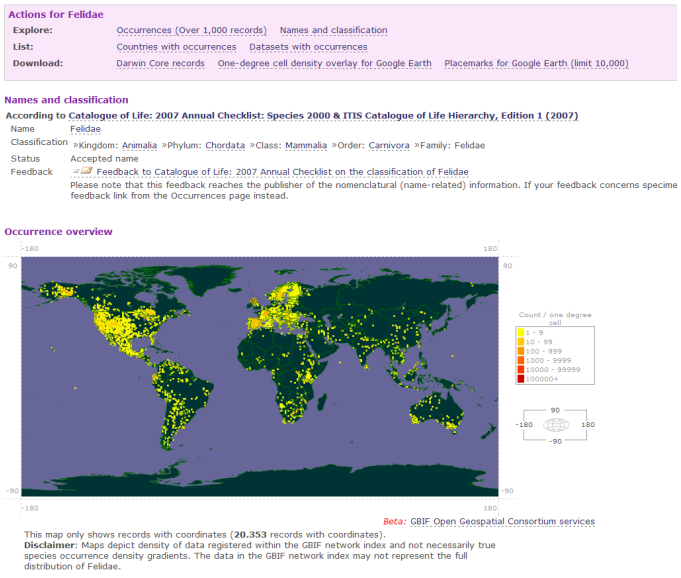

Figura 3.2: Ocorrências de resultados da família Felidae

número de fontes varia bastante, a disponibilidade dos dados também varia. Assim, o uso correto do portal de dados do GBIF é essencial para obter os resultados desejados.

As consultas no GBIF são similares às do Species 2000 e são realizadas sobre os nomes das espécies (científico ou comum) e níveis taxonômicos, além de ser possível procurar pelo nome de um país ou de uma base de dados, desde que esta esteja cadastrada no GBIF. O usuário pode utilizar diversos filtros para facilitar a busca, como encontrar as ocorrências de outras espécies na mesma localização, entre outros. Entretanto, estes filtros surgem como opção após uma busca inicial, que deve ser feita entre os 3 tipos citados. Além disso, o portal de dados permite que os resultados encontrados sejam visualizados em um mapa ou exportados no formato XML.

3.2.3 MaNIS- Mammal Networked Information System

O MaNIS é um portal para compartilhamento de bases de dados de espécimes, principalmente dados de museus. No sítio do portal[3], é possível escolher quais bases servirão de fonte para a consulta. Os critérios de busca são comuns a todas as espécies, não sendo possível selecionar algum atributo específico de alguma espécie. Os registros importados pelo sistema devem estar no formato Darwin Core.

Entre os objetivos do projeto do MaNIS, listados em (MANIS, 2009), está facilitar o acesso a dados de espécimes em navegadores, que é, em geral, um dos objetivos da maioria dos projetos pesquisados e também deste trabalho de conclusão. Assim como os outros sistemas, o MaNIS utiliza um protocolo já estabelecido para recuperação de

[3]http://manisnet.org/portals.html

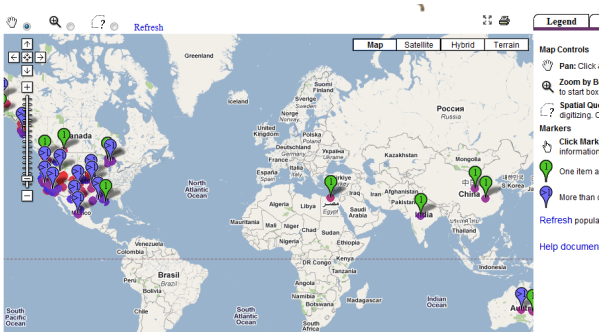

Figura 3.3: Resultado visual de uma consulta no MaNIS

informações, o Z39.50.

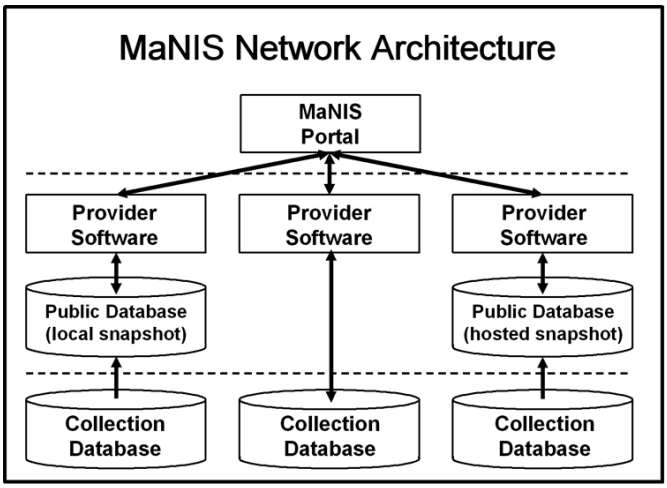

Figura 3.4: Estrutura do portal do MaNIS

Na figura 2.4, temos a estrutura do sistema do MaNIS mostrando o portal acessado pelo usuário no topo da estrutura. No meio da figura, temos os dados no padrão do MaNIS. As bases de dados das coleções são apresentadas na parte inferior da figura. As bases de dados das coleções estão localizadas fisicamente em cada museu, porém, os dados são exportados e o servidor do portal possui uma cópia de cada base no formato específico do MaNIS. Alterações feitas nas coleções dos museus são então replicadas nestas cópias.

3.2.4 Sistema Arctos

O Arctos[4] é um *site* que reúne dados de espécimes de coleções de museus, desenvolvido como um projeto no Museu de Zoologia de Vertebrados da Universidade de Berkeley, localizada na Califórnia (EUA). Como os outros portais já analisados, o objetivo do Arctos também é facilitar o acesso aos dados destas coleções, porém, ao contrário dos

[4]http://arctos.database.museum/home.cfm

outros, o sistema permite uma grande variedade de critérios de busca, como localização geográfica, características do espécime ou elemento taxonômico. Por exemplo, é possível procurar por todos os espécimes coletadas entre 1980 e 1990, da classe *Amphibia*, cujo método de conservação foi mumificação.

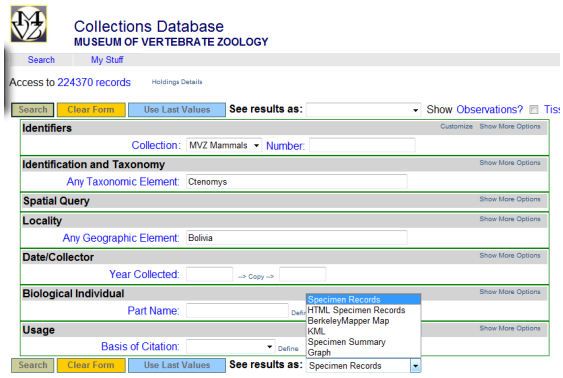

Figura 3.5: Exemplo de consulta no Arctos

Uma das principais funcionalidades do sistema é permitir que o usuário salve as consultas feitas no site para uso posterior, além de permitir que os resultados das consultas sejam exportados de diversas maneiras, como texto normal ou XML. O sistema permite que cada coleção tenha um funcionamento personalizado, permitindo que alguns critérios de busca sejam utilizados somente naquela coleção.

3.2.5 Sistema de Informação Ambiental do Biota

No Brasil, existe o programa BIOTA/FAPESP, formado com o *"objetivo de sistematizar a coleta, organizar e disseminar informações sobre a biodiversidade do Estado de São Paulo"* (BIOTA, 2009), que desenvolveu o Sinbiota - Sistema de Informação Ambiental do Biota. O Sinbiota, por sua vez, tem o objetivo de *"integrar informações geradas pelos pesquisadores vinculados ao Programa Biota/Fapesp e relacioná-las a uma base cartográfica digital de qualidade"* (SINBIOTA, 2009).

A partir do projeto Sinbiota foi desenvolvido o Atlas/Biota, um portal[5] para consultas sobre os dados coletados pelo Programa Biota. As consultas são textuais, sobre 7 campos: nome científico da espécie, autor da coleta, município, bacia hidrográfica, ecossistema, identificador interno e grupo taxonômico. Os resultados são exibidos em um mapa do estado de São Paulo, mas também é possível ver com detalhes as ocorrências, que incluem referências bibliográficas e uma breve descrição de como foi feita a coleta do espécime. Não existem dados sobre características das espécies, nem métodos para exportação dos dados disponíveis.

[5]http://www.biota.org.br/

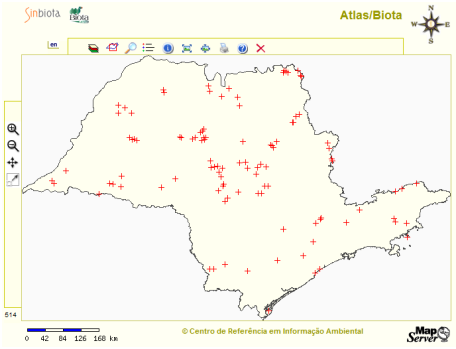

Figura 3.6: Ocorrências marcadas no mapa no Atlas/Biota

Como o nome Atlas sugere, o foco é na visualização dos resultados no mapa. Assim, é possível mostrar no mapa os rios do estado de São Paulo, representações da vegetação, delimitação de área urbana, entre outras opções. Após uma consulta inicial, é possível selecionar no mapa um subconjunto de ocorrências para visualização detalhada.

3.3 Comparação entre os sistemas estudados

Para comparar os sistemas estudados, foram definidos os seguintes critérios considerados relevantes para este projeto:

- navegação na árvore taxonômica - permite procurar diretamente na árvore e facilita a busca de algum nível específico pelo usuário;
- consultas visuais - permite a utilização de mapas para demarcar consultas geográficas;
- exportação de dados - permite a exportação dos resultados das consultas para uso em outra ferramenta é fundamental;
- possibilidade de consultas sobre características do espécime - este é um dos principais critérios de busca utilizados pelos usuários de sistemas de biodiversidade;
- e apresentação de bibliografia relacionada nos resultados.

Considerando a navegação na árvore taxonômica, o Species 2000 e o GBIF permitem que o usuário percorra os níveis taxonômicos existentes na base. No GBIF, é possível até mesmo salvar os resultados encontrados em um determinado nível no formato Darwin Core, por exemplo. No Arctos e no MaNIS é necessário realizar uma consulta para então saber se uma determinada família ou ordem existe na base.

O Arctos, o GBIF e o Atlas/Biota permitem que o usuário faça uma busca utilizando uma ferramenta visual. No caso do GBIF, a consulta é feita utilizando um plugin desenvolvido pela própria instituição, enquanto o Arctos utiliza o serviço do Google Maps. No Atlas, a ferramenta visual é restrita à seleção de um subconjunto dos resultados de uma consulta.

Com exceção do Species 2000 e do Atlas, que não têm como objetivo disponibilizar os dados, todos os outros sistemas oferecem a possibilidade de exportar dados, sejam como tabelas ou no formato XML. O MaNIS inclusive permite que o usuário escolha quais campos das tabelas devem ser incluídos no resultado da busca.

A consulta sobre características dos espécimes, como tamanho do crânio, só é permitida no Arctos. O GBIF e o MaNIS, por serem agregadores de dados, colocam como critérios de busca somente aqueles comuns a todas as espécies.

Por fim, o Species 2000, o Arctos e o Atlas apresentam nos resultados bibliografias associadas às espécies e aos espécimes, enquanto o GBIF e o MaNIS novamente deixam isto a cargo dos fornecedores dos dados.

Assim, a análise dos sistemas em função destes critérios resulta na tabela 2.1 apresentada a seguir.

Critério x Sistema	Species 2000	GBIF	MaNIS	Arctos	Atlas
Navegação na árvore taxonômica	Sim	Sim	Não	Não	Não
Consulta visual	Não	Sim	Não	Sim	Sim
Exportação de dados	Não	Sim	Sim	Sim	Não
Consulta sobre características	Não	Não	Não	Sim	Não
Resultados com bibliografia	Sim	Não	Não	Sim	Sim

3.4 Considerações finais

Comparando as funcionalidades dos sistemas estudados, vemos que o sistema Arctos é o que mais se aproxima da necessidade do projeto sobre o qual este trabalho foi desenvolvido. A busca sobre características dos espécimes é a principal função do trabalho e as opções oferecidas pelo Arctos são bastante similares ao que era desejado no projeto.

Outra função importante e comum a maioria dos sistemas é a possibilidade de exportar os resultados da busca. No sistema resultante do projeto deste trabalho, o usuário muitas vezes precisa dos dados em outro software. Portanto, é fundamental que o sistema seja otimizado para permitir isto de uma forma eficiente.

Em termos de interface, o GBIF é um modelo a ser seguido, pois mostra com clareza todas as opções disponíveis nos resultados da busca, além das funcionalidades citadas,

como visualização em mapa e exportação dos dados.

Ainda que nenhum sistema estudado apresente todas as características desejadas para o projeto deste trabalho, todos servem como referência para o desenvolvimento do módulo de consulta, que deve atender aos critérios selecionados o estudo.

4 ANÁLISE DE DADOS E O AMBIENTE R

Este capítulo descreve os principais conceitos de análise de dados do ponto de vista do banco de dados, o ambiente R, utilizado pelos biólogos participantes do projeto e as possíveis forma de integração do R com bases de dados de biodiversidade. Facilitar a análise de dados com o ambiente R é o principal objetivo da interface desenvolvida neste trabalho.

4.1 Análise de dados

Como o próprio nome sugere, um banco de dados tem a função de armazenar os dados para que sejam extraídos e manipulados em outra aplicação, seja um visualizador de banco de dados, um *site* ou um aplicativo comercial. Assim, originalmente, a linguagem SQL (*Structured Query Language*, em português: linguagem de consulta estruturada), que é utilizada para consultas em SGBD, não foi projetada para que fossem efetuadas operações mais complexas de análise. Ao longo dos anos, a SQL foi sendo alterada e, atualmente, existe suporte nativo a algumas funções de análise. Entretanto, estas funções normalmente não são otimizadas para um grande volume de dados.

Para suportar a análise de dados de forma eficiente, foram desenvolvidos os sistemas para processamento analítico *on-line* (OLAP, de *on-line analyitical processing*). Os primeiros sistemas OLAP armazenavam na memória do computador cubos de dados resumidos, resultantes da construção de tabelas pivô. Estas tabelas também são chamadas de tabulações cruzadas, na qual se relacionam normalmente 3 ou mais atributos e cruzam-se os valores destes atributos. Em uma base de dados de biodiversidade, por exemplo, poderia ser criada uma tabela onde fosse armazenado o peso médio dos espécimes para todas as alturas e latitudes encontradas na base.

Em um sistema com muitos atributos e tabelas, o espaço ocupado no disco deve crescer rapidamente. Por isso, com o passar do tempo, os sistemas OLAP foram modificados para armazenar os dados resumidos em tabelas no próprio banco, recebendo a

denominação de OLAP relacional (ROLAP) (SILBERSCHATZ, 2006). Também foram desenvolvidos sistemas híbridos, que armazenam parte das tabelas em memória e outra parte no banco de dados, chamados de OLAP híbrido (HOLAP).

No entanto, este método resulta em possivelmente dezenas de tabelas para muitos atributos, além de ser ineficiente quando estes atributos variam. Para um projeto como o *TaxonomyBrowser*, que pode armazenar milhares de espécimes com atributos diferentes, um sistema OLAP se torna inviável. Porém, como a maior parte dos SGBD permite a extensão dos comandos básicos do SQL, ainda é possível integrar a análise com o armazenamento dos dados.

4.2 O ambiente R

O ambiente R, utilizado pelos biólogos participantes do projeto, inclui uma linguagem de programação e uma interface de linha de comando e é considerado um ambiente padrão *de fato* para desenvolvimento de *softwares* de estatística (FOX; ANDERSEN, 2005). O projeto do ambiente R foi tornado público em 1993, sendo desenvolvido inicialmente por dois pesquisadores do departamento de estatísticas da *University of Auckland* (IHAKA, 1998). A linguagem utilizada no ambiente R é bastante similar a *S*, que foi por muito tempo linguagem padrão para análise estatística, enquanto o funcionamento do sistema deriva da linguagem *Scheme*. Logo após o lançamento, o código-fonte do sistema foi disponibilizado com a licença *GNU General Public License*, que resultou no desenvolvimento de centenas de módulos de diversas áreas do conhecimento: biologia, computação, matemática, física, economia, entre outras. A maior parte dos módulos está disponível no *Comprehensive R Archive Network* (CRAN)[1], que classifica os módulos de acordo com sua categoria. Além disso, o ambiente R faz o *download* e a instalação automática a partir do CRAN se um script necessitar de um módulo que não esteja instalado.

Na linguagem R, qualquer dado é tratado como um objeto. O programador pode declarar números, cadeias de caracteres, vetores, *arrays*, entre outros; todos eles se tornam objetos na sessão que está sendo executada. Para a análise de dados, o principal tipo de objeto é o *data frame*, que consiste em uma espécie de tabela. O *data frame* armazena as linhas da tabela, na qual cada uma contém um par de chave e valor. É possível construir um *data frame* a partir de um arquivo ou inserir linha por linha no *data frame* com o comando *rbind*.

Por ser um *software* desenvolvido para análise estatística, o ambiente R oferece funções para modelagem e geração de gráficos, de *boxplots* a histogramas. Os gráficos podem ser salvos em PDF, PS, PNG e outros formatos conhecidos. Se o programador tiver experiên-

[1] http://cran.r-project.org/

cia com a linguagem, pode inclusive desenhar seus próprios gráficos utilizando comandos de baixo nível da linguagem R.

4.3 Integração do ambiente R com outros sistemas

Para integrar o ambiente R com o *TaxonomyBrowser*, foram estudados os principais métodos para que isso fosse possível. Primeiramente, foi encontrada a PL/R (*Procedural language R*, isto é, linguagem procedural R), uma extensão para o SGBD *PostgreSQL* que permite que o usuário do SGBD escreva funções na linguagem R como complemento ao SQL. Quando a PL/R é utilizada, não há a necessidade de exportar os dados para uma aplicação de análise de dados, diminuindo a quantidade de recursos ocupados no sistema. Um usuário com experiência em R e SQL pode implementar funções complexas para a análise de dados, enquanto outros usuários menos familiarizados com a linguagem podem simplesmente utilizar estas funções. Entretanto, esta opção está limitada a um determinado SGBD que não é o utilizado pelo *TaxonomyBrowser*. Mesmo assim, caso o *TaxonomyBrowser* seja alterado para utilizar *PostgreSQL*, a PL/R é uma boa opção para melhorar a integração.

No nível da aplicação, acima do banco de dados, as formas de integração variam entre as linguagens de programação e sistema operacional. Para a linguagem Java, por exemplo, existe a biblioteca JRI (*Java/R interface*), que permite executar código da linguagem R em um aplicativo Java. Para PHP, existe uma alternativa similar, o R-php. O R-php funciona como uma interface entre o usuário e o ambiente R, permitindo que o usuário acesse uma página *web*, insira comandos na linguagem R e receba o resultado na tela. Desta forma, não é uma alternativa viável para ser utilizada.

No escopo do sistema operacional, existe o conceito de comunicação entre processos (IPC, de *inter-process communication*). A comunicação entre processos ocorre quando um ou mais processos necessitam coordenar suas atividades, por exemplo, ao acessar um determinado recurso. (GRAY, 2003). Entre as soluções de IPC implementadas em sistemas UNIX, a que melhor atende as necessidades da integração do *TaxonomyBrowser* com o ambiente R é o canal nomeado (*named pipe*), que será descrito no próximo capítulo.

5 INTERFACE DE CONSULTAS ANALÍTICAS PARA O *TAXONOMYBROWSER*

Este capítulo descreve a Interface de Consultas Analíticas desenvolvida para o *TaxonomyBrowser*. A implementação da interface foi feita considerando que o portal foi desenvolvido utilizando a linguagem PHP e que os usuários do portal utilizam o ambiente R para análise de dados. Assim, a interface foi projetada com o objetivo de utilizar o que estivesse disponível para a integração tanto no lado do portal como no lado do ambiente R.

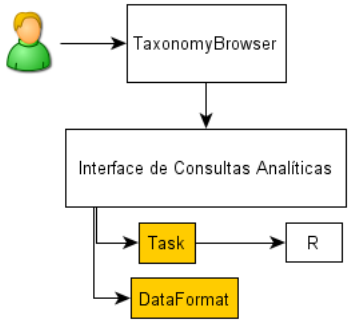

Figura 5.1: Componentes da interface

A figura 5.1 mostra os componentes da interface na arquitetura do *TaxonomyBrowser*. Para o usuário, não há distinção entre as interfaces do sistema, porém, internamente a interface de consultas analíticas executa o R através da classe *Task*, após a conversão dos dados para o formato adequado de execução no R feita pela classe *DataFormat*.

5.1 Criação de processos em PHP

A principal função da interface é enviar os dados existentes no banco de dados e transferi-los a um *script* na linguagem R para que sejam analisados. O usuário acessa o portal de dados, realiza uma busca, escolhe um *script* com o qual os dados serão analisa-

dos e depois recebe o resultado na tela, normalmente uma imagem. A implementação em PHP da interface é responsável pela execução do sistema R e pela transferência dos dados. Além de funções de PHP que executam programas com passagem simples de argumento, existe a função *proc_open()*, que permite executar um processo com variáveis de ambiente exclusivas para o processo, como, por exemplo, caminhos de diretórios. Além disso, através desta função é possível definir dois *pipes* para comunicação entre a página PHP e o programa executado. Para esta funcionalidade da interface, foi desenvolvida uma classe *Task*, responsável pela execução do processo e envio e recebimento de mensagens através do *pipe*, bem como a verificação do estado do processo.

5.2 Comunicação entre processos e pipes

Como mencionado, um dos conceitos de comunicação entre processos é o canal, ou *pipe*, que é um recurso criado no sistema operacional compartilhado entre dois processos. O *pipe* é utilizado quando é necessário que um programa envie mensagens para outro, como, por exemplo, um sinal para que um dos programas seja encerrado. A forma mais básica de um *pipe* ocorre em terminais UNIX utilizando o comando "para que um processo envie dados para outro. Este canal criado é também chamado de *unnamed pipe* (canal sem nome), pois é temporário e aberto pelo terminal (também chamado de *shell*) enquanto os processos são executados. Quando a execução termina, o canal é fechado. Um *unnamed pipe* é geralmente utilizado quando um programa gera dados de saída e outro programa recebe dados para a entrada. O *pipe* então serve como o canal pelo qual um programa envia estes dados para o outro.

Esta forma de comunicação é insuficiente para programas que são executados ao mesmo tempo e, principalmente, quando a execução de um depende de outro. Neste caso, utiliza-se um *named pipe* (canal nomeado). Em sistemas UNIX, um *named pipe* é um canal criado no sistema de arquivos do sistema operacional e tratado como se fosse um arquivo, incluindo permissões de leitura e escrita. Este canal funciona como uma fila de mensagens FIFO (*first in, first out*, em português: primeiro a entrar, primeiro a sair). Um processo com a permissão adequada pode enviar mensagens por este canal e outro pode "ler"o canal como estivesse lendo um arquivo e receber as mensagens. Quando é necessário que um processo envie uma mensagem de resposta, cria-se outro *pipe*. Assim, são abertos dois canais de comunicação entre os processos, com um sentido diferente de envio de mensagens em cada um deles.

5.3 Formatação de dados para a linguagem R

Após a criação do processo que executará o *script* R, é necessário enviar os dados do banco. A linguagem R possui uma sintaxe própria, portanto, a interface deve formatar os dados antes de passá-los para o *script*. Por outro lado, o *script* deve ser programado de forma que possa manipular os dados oriundos do banco de dados. Isto significa seguir um padrão definido para quem utilizar a interface em relação à nomenclatura dos dados. Por exemplo, um *script* deve esperar que coordenadas geográficas sejam chamadas de *Longitude* e *Latitude*, com as letras iniciais em maiúsculo, ou que *tamanho do crânio* seja enviado pelo *script* como *TamanhoDoCranio*. Como os *scripts* são executados em um programa externo e enviados pelos próprios usuários para que sejam utilizados no portal, foi definido que a responsabilidade pelo funcionamento correto de cada *script* é do autor do mesmo.

A interface, além de executar o interpretador R, envia os dados e os parâmetros básicos para a execução de um *script*. Esta execução é feita simulando o uso do interpretador por um usuário, em que o programa PHP envia através do *pipe* os comandos para inserção dos dados no ambiente do interpretador R. Inicialmente, a interface deve importar os dados da base. Cada pesquisa realizada por um usuário é salva no portal e é recuperada quando ele deseja analisar os dados resultantes. Para importar os dados, a interface lê um arquivo em formato XML (*extensible markup language*, em português: linguagem de marcação estendível) que contém os resultados da busca. Isto é feito através da classe *DataFormat*, a qual percorre o conjunto de dados do arquivo e formata-os em strings (cadeias de caracteres) que contêm os comandos para a construção de *data frames* na linguagem R, descritos no capítulo anterior.

```
1 data ← NULL;
2 data ← rbind(data,data.frame("size"= 1111.21,"Latitude"= 29.92,"Dip"= "48n"));
3 ...;
4 session ← "images/2";
5 source("scripts/script_selecionado.r");
```

Algoritmo 1: Exemplo

O primeiro passo para a execução do *script* é a inicialização de um *data frame* vazio, através do comando da linha 1 no algoritmo 1. A seguir, o objeto do tipo DataFormat itera sobre seus dados e envia, linha por linha, o comando para construção do *data frame*, como exemplificado na linha 2. Por fim, antes que seja feita a análise dos dados, também é criada uma variável que contém o nome da imagem gerada, na linha 3. Após o envio dos parâmetros iniciais, o comando da linha 4 é enviado pelo *pipe*. O comando *source* carrega e executa do disco um *script* em R. Como dito, é esperado que este *script* acesse as variáveis seguindo o padrão de nomes: a variável *data* contém os dados e a variável *name* contém o nome do arquivo da imagem. O *script* carregado também deve escrever

na tela uma mensagem sinalizando o final da execução.

Como o interpretador R foi executado a partir da função proc_open(), com a criação de um *unnamed pipe* para leitura por parte do PHP, qualquer texto gerado pelo interpretador sera enviado através deste *pipe*. Portanto, quando a página PHP termina a montagem do *script*, passa automaticamente a um estado de espera até que receba esta sinalização. Caso seja ultrapassado um limite de tempo estabelecido, a página irá interromper a espera, verificar se a imagem resultante foi criada e, caso o arquivo não exista, informará o usuário que ocorreu um erro durante a execução.

5.4 Exemplo de execução

As figuras 2, 3 e 4 mostram um exemplo do uso da interface. A figura 2 mostra a tela de visualização de uma busca no *TaxonomyBrowser*, quando o usuário tem permissão para fazer uma análise sobre os resultados da busca.

Ao clicar no botão *Export*, o usuário é redirecionado para a tela exibida na figura 3. Nesta tela, o usuário visualiza a tabela com o resultado da busca e pode selecionar um *script* R para ser executado.

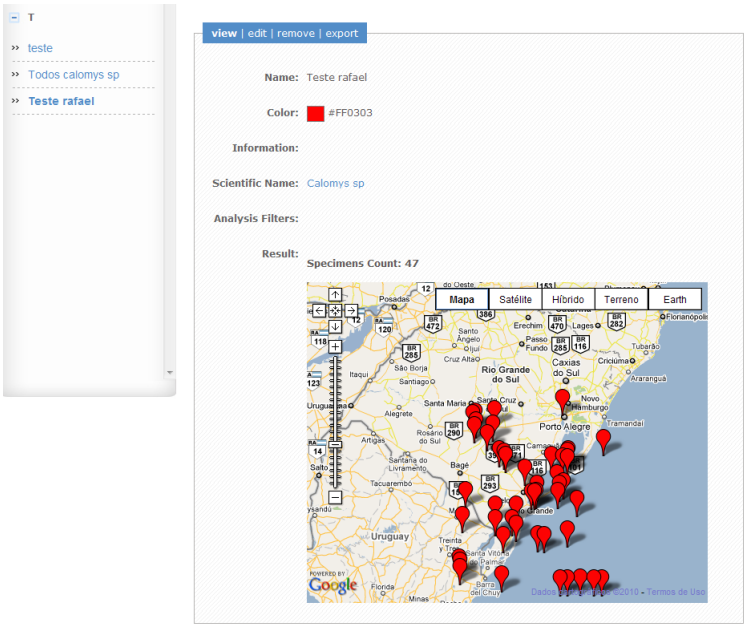

Figura 5.2: Visualização de busca no *TaxonomyBrowser*

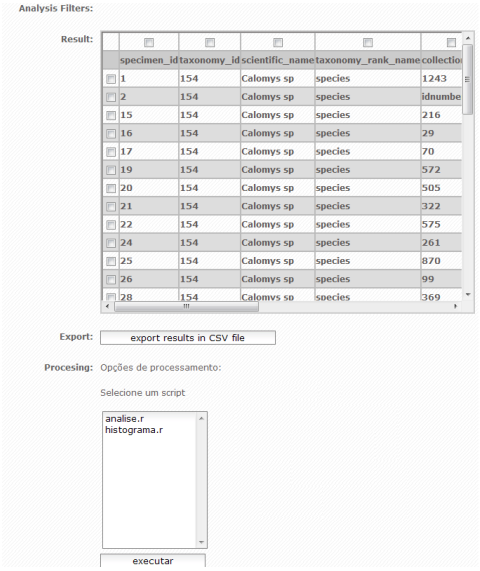

Figura 5.3: Seleção de *script*

Por fim, se o *script* foi executado com sucesso, o usuário recebe a imagem resultante deste *script*, como mostrado na figura 4.

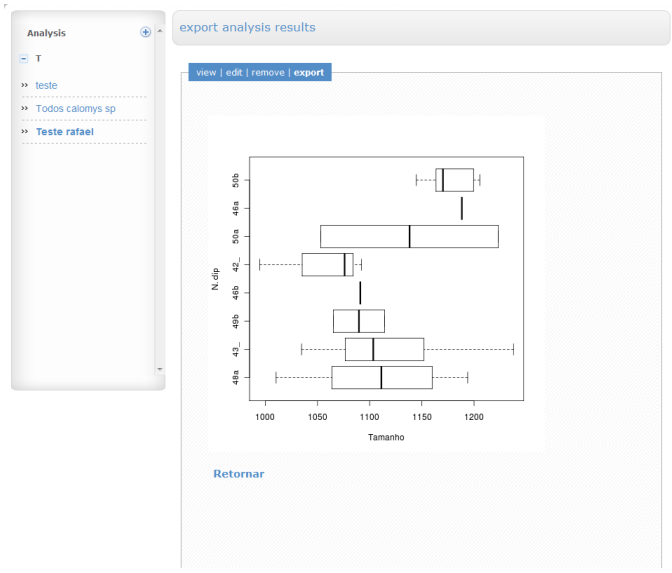

Figura 5.4: Resultado do *script*

5.5 Avaliação preliminar da interface

Ao analisar as vantagens e desvantagens da implementação da interface, é preciso considerar a relevância de alguns aspectos. Por exemplo, em um sistema acessado por muitos usuários, a criação de um processo do R para cada usuário poderia facilmente causar uma sobrecarga no servidor. No entanto, no *TaxonomyBrowser* isto não deverá ser um problema, pois o número de usuários será pequeno.

A principal desvantagem da implementação está no fato do ambiente R salvar as imagens geradas em arquivo. Em uma linguagem de programação de baixo nível, normalmente é possível acessar os *bytes* desta imagem. No R, entretanto, ao escolher um arquivo como alvo da geração de um gráfico, não há forma de capturar estes *bytes*. Assim, apesar da interação entre o PHP e o R ser feita através do canal nomeado, o resultado final da análise não é retornado ao PHP, mas salvo no disco, fora do controle direto do PHP. A interface verifica se o arquivo foi criado, como esperado, mas não pode garantir que o arquivo foi criado. Isto fica a cargo do *script*, que é o próximo ponto avaliado.

O fato do *script* R não fazer parte do *TaxonomyBrowser* é uma vantagem para os usuários e uma desvantagem para os administradores. Novos *scripts* podem ser desenvolvidos sem a dependência do portal e os usuários podem facilmente corrigir problemas em *scripts* disponibilizados no *TaxonomyBrowser*. Para os administradores, a ocorrência de um problema com algum *script* torna difícil descobrir onde está o problema: no *script* ou no portal. Além disso, o administrador não tem como prever o que irá acontecer com a execução do *script*. Mesmo que o *upload* de um novo *script* seja verificado por um administrador, o usuário pode enviar uma nova versão de um *script* e pode provocar problemas no servidor ou somente ter um *script* que não funcione, deixando outros usuários na dependência tanto do administrador como do autor do *script* para a correção de erros.

A maior vantagem da separação entre a interface e o programa utilizado para análise, entretanto, está na possibilidade de utilização de outro *software* além do R. Se em algum momento o R deixar de ser a opção principal para análise dos dados ou surgir alguma outra ferramenta útil, a interface pode ser facilmente alterada para adotar a outra ferramenta. Caso a integração ocorresse na camada do banco de dados, não haveria outra opção que não fosse desenvolver uma nova interface na camada da aplicação. Além disso, na forma atual, a interface funciona efetivamente como um *componente* do sistema. Procedimentos de manutenção podem ser feitos sem a interrupção do servidor e o funcionamento geral do *TaxonomyBrowser* independe da interface.

Por fim, existe uma desvantagem que ocorreria de qualquer forma em uma solução na camada da aplicação, que é a restrição de uma imagem por *script* executado. As análises se tornam restritas àquelas que obrigatoriamente resultam em um gráfico, excluindo análises mais simples como medidas de média e variância. Resultados numéricos,

portanto, devem ser exibidos na forma de um gráfico, ainda que isto seja ineficiente. Esta regra do sistema poderá ser flexibilizada no futuro com o constante desenvolvimento do *TaxonomyBrowser*, mas foi estabelecida para o funcionamento correto até o final deste trabalho.

6 CONCLUSÕES

Este trabalho descreve uma interface que permite aos usuários do *TaxonomyBrowser*, um portal facilitador de acesso a dados de biodiversidade, realizar análises estatísticas sobre os dados cadastrados no sistema, dadas as características apresentadas pelo portal:

- SGBD *MySQL*
- Portal desenvolvido com a linguagem *PHP*
- Análise estatística realizada através do ambiente R.

Para atingir este objetivo, foram estudados os principais portais de dados de biodiversidade e métodos para a integração de análise e armazenamento de dados, levando em consideração as características técnicas acima listadas. A implementação da interface se deu através da comunicação entre processos, um conceito da área de sistemas operacionais amplamente utilizado desde tarefas mais básicas até interações complexas entre *softwares*.

O uso de canais nomeados permitiu que a interface permanecesse independente do *software* de análise R, possibilitando o uso de outros programas no futuro. Esta solução também apresentou a restrição de uma imagem por *script* executado e a impossibilidade de exibir resultados numéricos de forma textual. Como o controle dos *scripts* não é feito diretamente pela interface, esta regra foi imposta para garantir uma consistência destes *scripts*. Assim, assume-se que todos os *scripts* enviados para o portal sempre tem o mesmo resultado: uma imagem.

Por parte de quem utilizará o portal, a interface apresenta uma grande facilidade para trabalhos de campo. Apesar do poder de processamento dos computadores ter aumentado consideravelmente nos últimos anos, aparelhos portáteis ainda não apresentam um desempenho satisfatório para cálculos demorados. A coleta e armazenamento de dados no *TaxonomyBrowser* permite que o usuário com um computador portátil possa realizar

análises sobre dados recém coletados no *site*, sem a necessidade de ter o ambiente *R* instalado em seu computador. Além disso, até mesmo usuários que não tem familiaridade com linguagens de programação e *softwares* de análise estatística podem executar *scripts*.

Apesar deste trabalho estar encerrado, o desenvolvimento do *TaxonomyBrowser* continuará, com o início do uso "público" do portal e o armazenamento de muitos dados coletados que no momento estão desorganizados. Para a interface, futuramente, é de se esperar uma flexibilização da regra de que um script deve gerar somente uma imagem, se o uso do portal ocorrer como esperado. Medidas de ocupação de espaço de disco e tempo para execução com análises complexas só poderão ser feitas após a inclusão de uma grande quantidade de dados, o que também só irá ocorrer no futuro.

REFERÊNCIAS

Programa Biota/Fapesp. Disponível em: http://www.biota.org.br/info/index. Acesso em: nov. 2009.

CAÑETE, S. C. et al. **Integrando visualização e análise de dados em sistema de gerenciamento de dados de biodiversidade**. IV e-Science Workshop, jul. 2010.

TaxonomyBrowser. Disponível em: http://darwin.inf.ufrgs.br/taxonomybrowser. Acesso em: jul. 2010.

FOX, J.; ANDERSEN, R. **Using the R Statistical Computing Environment to Teach Social Statistics Courses**. Hamilton, Ontario, Canada. Disponível em: www.unt.edu/rss/Teaching-with-R.pdf. Acesso em: jul. 2010.

Global Biodiversity Information Facility. Disponível em: http://www.gbif.org. Acesso em: nov. 2009.

GRAY, J. S. **Interprocess communications in Linux**. Upper Saddle River: Prentice Hall, 2003.

IHAKA, R. **R : Past and Future History**. Auckland, New Zealand. Disponível em: http://cran.r-project.org/doc/html/interface98-paper/paper.html. Acesso em: jul. 2010.

JONES, A. C. SPICE: A Flexible Architecture for Integrating Autonomous Databases to Comprise a Distributed Catalogue of Life. In: International, Conference and Workshop on Database and Expert Systems Applications, 11. **Proceedings...**. Springer Berlin, (Lecture Notes in Computer Science), p.981-992, 2000.

Mammal Networked Information System. Disponível em: http://manisnet.org. Acesso em: nov. 2009.

SILBERSCHATZ, A. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Elsevier, 2006.

SinBiota - Sistema de Informação Ambiental do Biota. Disponível em: http://sinbiota.cria.org.br/. Acesso em: nov. 2009.

SPICE Software. Disponível em: http://www.sp2000.org/index.php?option=com_content&task=view&id=38&Itemid=49. Acesso em: nov. 2009.