

Dissertação de Mestrado

**Ambiente computacional para suporte a
“*Reproducible Research*”**

Rodrigo de Lima Pinheiro
rodrigolpinheiro@gmail.com

Orientador:
Alejandro C. Frery

Maceió, Julho de 2013

Rodrigo de Lima Pinheiro

**Ambiente computacional para suporte a
*“Reproducible Research”***

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Orientador: Alejandro C. Frery

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecária Responsável: Helena Cristina Pimentel do Vale - CRB4 - 661

P654d Pinheiro, Rodrigo de Lima.

Ambiente computacional para suporte a *Reproducible Research* /
Rodrigo de Lima Pinheiro. – 2018

77 f. : il.

Orientador: Alejandro C. Frery.

Dissertação (Mestrado em Modelagem Computacional de
Conhecimento) – Universidade Federal de Alagoas. Instituto de
Computação. Maceió, 2013.

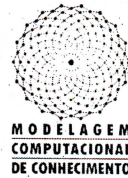
Bibliografia: f. 76–77.

1. Reproducible Research. 2. Software FLOSS. 3. Pesquisa
Computacional Quantitativa. I. Título.

CDU: 004.4



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364



Membros da Comissão Julgadora da Dissertação de Mestrado de Rodrigo de Lima Pinheiro, intitulada: “Ambiente computacional para suporte a “Reproducible Research””, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 31 de julho de 2013, às 14h30min, na Sala de Treinamento do Laboratório de Computação Científica e Visualização (LCCV).

COMISSÃO JULGADORA

Prof. Dr. Alejandro Cesar Frery

UFAL – Instituto de Computação

Orientador

Prof. Dr. Eduardo Nobre Lages

UFAL – Centro de Tecnologia

Examinador

Prof. Dra. Fátima Nelsizeuma Sombra de Medeiros

UFC – Departamento de Engenharia de Teleinformática

Examinadora

Maceió, julho de 2013.

RESUMO

Reproducible Research (RR) define boas práticas em metodologia da pesquisa científica para serem aplicadas, em especial à pesquisa computacional quantitativa. Diversos periódicos de prestígio internacional adotam ou exigem o cumprimento dessas práticas. Entretanto, o pleno atendimento às premissas de RR implica em uma carga extra de trabalho. Tal fato compromete seriamente a sua adoção, sendo desejável a utilização de ferramentas computacionais que minimizem esse esforço. Como forma de garantir que as práticas de RR sejam aplicadas de forma transparente a artigos de pesquisa computacional quantitativa, apresentam-se neste trabalho: modelagem, proposta e construção de um ambiente computacional desenvolvido a partir da integração de ferramentas computacionais *Free/Libre and Open-Source Software* (FLOSS). Tal ambiente computacional é implementado com base na interseção do resultado de três frentes de trabalho: o estudo das principais práticas de *Reproducible Research*; a análise de soluções computacionais de suporte a RR voltadas especificamente para ciências computacionais; e o levantamento e integração de softwares FLOSS para suporte sustentável a esta iniciativa.

Palavras-chave: reproducible research. software FLOSS. pesquisa computacional quantitativa.

ABSTRACT

Reproducible Research (RR) defines good practices in scientific research methodology to be applied in special to quantitative computational research. Several prestigious international journals adopt or require the compliance of these practices. However, the full compliance of RR premises implies in extra workload. Such fact seriously compromises its adoption, then it is preferable the usage of computer tools that will minimize this efforts. In order to ensure that the RR practices are transparently applied to papers of quantitative computational research, this paper presents: the modeling, proposal and the development of a computational environment that was developed by the integration of Free/Libre and Open-Source Software (FLOSS) tools. Such computing environment has been implemented based on the intersection of the result of three work fronts: study of the main practices of Reproducible Research, analysis of computational solutions to support RR specifically to computer sciences; survey and FLOSS software integration for sustainable support to this initiative.

AGRADECIMENTOS

Quero aproveitar a oportunidade para registrar as condições sob as quais finalizei esse trabalho. Ao mesmo tempo em que escrevo essa dissertação, também tenho que atender as demandas do Núcleo de Tecnologia da Informação (NTI), do Laboratório de Computação Científica e Visualização (LCCV) e da construção de uma casa. Para que todas essas atividades ocorram é necessário o engajamento de muitas pessoas, e é para elas que eu deixo os meus mais sinceros agradecimentos.

A minha noiva Thayse Fonseca, que assumiu grande parte das demandas da construção da casa, além de ajudar com as questões relacionadas às correções de português desse trabalho.

A minha família, por ter me incentivado durante todo o período do mestrado.

Ao meu orientador Alejandro C. Frery, que contribuiu mostrando o passo a passo da vida de um pesquisador.

Aos meus colegas do NTI e do LCCV, por terem “feito” meu trabalho nos momentos das minhas faltas e impedimentos.

Rodrigo de Lima Pinheiro

LISTA DE ILUSTRAÇÕES

Figura 2.1	Espectro da reprodutibilidade	18
Figura 4.1	Casos de uso do ambiente computacional	35
Figura 4.2	Modelo Entidade-Relacionamento do ambiente computacional	38
Figura 4.3	Modelo lógico do ambiente computacional	39
Figura 4.4	Camadas do ambiente computacional	41
Figura 4.5	Diagrama de classes do pacote <i>domain</i>	42
Figura 4.6	Diagrama de classes do pacote <i>persistence</i>	44
Figura 4.7	Diagrama de classes do pacote <i>service</i>	46
Figura 4.8	Diagrama de atividades relacionado ao login no ambiente	47
Figura 4.9	Diagrama de atividades relacionado à gestão de pesquisadores e colaboradores.	48
Figura 4.10	Diagrama de atividades relacionado à gestão de projetos de pesquisa	49
Figura 4.11	Diagrama de atividades relacionado à gestão de versão de um projeto de pesquisa	50
Figura 4.12	Diagrama de atividades relacionado à gestão da configuração do ambiente	51
Figura 4.13	Diagrama de atividades relacionado à gestão de artefatos do projeto de pesquisa	52
Figura 4.14	Diagrama de atividades relacionado à geração da página web de um projeto de pesquisa	53
Figura 5.1	Tela de login do ambiente computacional	56
Figura 5.2	Tela inicial do usuário logado	57
Figura 5.3	Tela de erro no login do ambiente computacional	57
Figura 5.4	Tela com a listagem dos pesquisadores cadastrados no ambiente computacional	58
Figura 5.5	Tela que exibe os detalhes do cadastro de um pesquisador no ambiente computacional	59
Figura 5.6	Tela de cadastro do pesquisador	60
Figura 5.7	Tela com a listagem de projetos associados ao pesquisador/colaborador logado	61
Figura 5.8	Tela de visualização dos dados básicos do projeto de pesquisa	62
Figura 5.9	Tela de visualização das configurações do projeto de pesquisa	62
Figura 5.10	Tela de visualização da associação entre pesquisadores e projeto	63
Figura 5.11	Tela de edição dos dados básicos do projeto de pesquisa	64
Figura 5.12	Tela de edição dos dados de configuração do projeto de pesquisa	64
Figura 5.13	Tela de edição da associação entre pesquisadores e um projeto de pesquisa	65
Figura 5.14	Tela com a opção de criação de um novo projeto de pesquisa	66
Figura 5.15	Tela de geração da nova versão do projeto de pesquisa	67
Figura 5.16	Botão de confirmação para geração de uma nova versão	67
Figura 5.17	Tela de visualização da associação entre pesquisadores e projeto	68
Figura 5.18	Tela de gestão dos artefatos de um projeto de pesquisa	69
Figura 5.19	Tela de geração da página web para o projeto de pesquisa	70
Figura 5.20	Botão de confirmação para geração de uma nova página web para o projeto de pesquisa	70

Figura 5.21 Site de apoio ao projeto	71
Figura 5.22 Publicações originadas a partir do desenvolvimento do ambiente computacional	72
Figura 5.23 Exemplo de artefatos disponibilizados em cada publicação	72
Figura 5.24 Exemplo de artefatos disponibilizados para o ambiente computacional	73

LISTA DE TABELAS

Tabela 2.1	Quadro comparativo de soluções	22
Tabela 3.1	Mapeamento das principais funcionalidades do ambiente	23
Tabela 3.2	Softwares escolhidos para o ambiente computacional	31

SUMÁRIO

1	Introdução	8
1.1	Introdução	8
1.2	Motivação	9
1.3	Solução proposta	11
1.4	Objetivos	12
1.5	Metodologia	12
1.6	Contribuições	13
2	Conceitos e iniciativas de <i>Reproducible Research</i> na ciência computacional	14
2.1	Práticas de <i>Reproducible Research</i>	14
2.2	Vantagens e desvantagens da utilização de <i>Reproducible Research</i>	16
2.3	<i>Reproducible Research</i> na ciência computacional quantitativa	17
2.3.1	Análise de soluções computacionais	18
3	Requisitos e proposta do ambiente computacional	23
3.1	Requisitos para um ambiente computacional para suporte a <i>Reproducible Research</i>	23
3.2	Proposta para o ambiente computacional	24
3.3	Requisitos adicionais para o ambiente computacional	26
3.4	Softwares FLOSS componentes do ambiente	27
3.4.1	Gestão de página web	28
3.4.2	Gestão de versões de artefatos	29
3.4.3	Gestão de fóruns	30
3.4.4	Gestão de armazenamento de dados	30
3.4.5	Softwares selecionados	31
4	Modelagem do ambiente computacional proposto	32
4.1	Tecnologias e ferramentas utilizadas	32
4.1.1	Linguagem de programação <i>Java</i>	32
4.1.2	Mapeamento objeto-relacional <i>Hibernate</i>	33
4.1.3	Interface web para o usuário <i>Java Server Faces</i>	33
4.1.4	Ambiente de desenvolvimento integrado <i>NetBeans</i>	33
4.1.5	Servidor web <i>Apache Tomcat</i>	34
4.1.6	Modelagem UML <i>Jude</i>	34
4.1.7	Modelagem de banco de dados relacional <i>DIA</i>	34
4.2	Modelagem do ambiente computacional	34
4.2.1	Requisitos funcionais	34
4.2.2	Modelagem da base de dados	37
4.3	Projeto do ambiente computacional	40
4.3.1	Arquitetura do ambiente computacional	40
4.3.2	Diagrama de classes	41
4.3.3	Diagrama de atividades	47
5	Ambiente computacional proposto	54

5.1	Preparação do ambiente computacional	54
5.1.1	Instalação e configuração do ambiente computacional	54
5.2	Principais funcionalidades do ambiente computacional	55
5.2.1	Efetuar login	55
5.2.2	Gestão de pesquisador/colaborador	58
5.2.3	Gestão de projeto de pesquisa	60
5.2.4	Gestão de versão de um projeto de pesquisa	66
5.2.5	Gestão de configuração do ambiente	68
5.2.6	Gestão de artefatos do projeto de pesquisa	68
5.2.7	Geração da página web de um projeto de pesquisa	69
5.3	Projeto reprodutível	71
6	Considerações finais	74
6.1	Conclusão	74
6.2	Trabalhos futuros	75
	REFERÊNCIAS	76

1

Introdução

Segundo Michel Blay, a ciência é “o conhecimento claro e evidente de algo, fundado quer sobre princípios evidentes e demonstrações, quer sobre raciocínios experimentais, ou ainda sobre a análise das sociedades e dos fatos humanos”. A ciência não se considera “dona” da verdade absoluta e inquestionável. A partir do racionalismo crítico, todas as suas verdades podem ser quebradas, bastando apenas um mínimo de evidência.

1.1 Introdução

Há cerca de 20 anos, Jon Claerbout, um pesquisador da Universidade de *Stanford*, que trabalha com análise de dados e desenvolvimento de algoritmos para a exploração geofísica, relatou que possuía grande dificuldade para reproduzir algoritmos e resultados relatados por outros pesquisadores (Donoho et al., 2009). Como agravante, a mesma situação repetia-se em seu laboratório, ou seja, com bastante frequência seus alunos e pesquisadores enfrentavam problemas para reproduzir os resultados gerados pelas próprias pesquisas (Schwab et al., 2000).

Esse fato “provocou” a equipe de Claerbout a desenvolver um sistema de documentação para reprodução dos resultados em computação científica ¹, que passou a ser utilizado pelos colaboradores do seu laboratório. Tal software inspirou a implementação de várias ferramentas que auxiliam a reprodução dos resultados de um artigo em pesquisa computacional.

¹Maiores detalhes podem ser encontrados na Seção 2.3.1

Assim, o termo *Reproducible Research* (RR) foi criado no contexto da pesquisa computacional (Fomel & Claerbout, 2009). *Reproducible Research* é uma sistematização de boas práticas em metodologia da pesquisa científica, especialmente organizadas para a pesquisa computacional quantitativa.

As práticas recomendadas por RR são sugeridas, e até mesmo exigidas, por veículos internacionais de pesquisa. No entanto, como se tratam de ações que não fazem parte do trabalho cotidiano do pesquisador, a sua adoção é dificultada. Como forma de auxiliar os pesquisadores a atender os requisitos de RR, apresenta-se neste trabalho um ambiente computacional que possui como principal objetivo a aplicação das práticas de RR em um projeto de pesquisa computacional quantitativa.

Nos capítulos e seções a seguir, serão apresentadas as práticas de *Reproducible Research*, vantagens e desvantagens da sua utilização na pesquisa computacional quantitativa, bem como as principais iniciativas em termos de soluções computacionais encontradas na literatura. Por fim, serão mostrados os requisitos, a proposta, modelagem, e as principais funcionalidades do ambiente computacional.

1.2 Motivação

Na pesquisa computacional quantitativa é bastante comum encontrar artigos que, em suma, transformam dados brutos em resultados computacionais a serem publicados. Para isso os pesquisadores fazem uso de softwares, que em muitos casos são desenvolvidos apenas para solucionar o problema desejado. Em uma análise apurada, percebem-se diversos problemas no código fonte desses softwares. Entre os principais problemas enumeramos: falta de documentação, poucos ou nenhum comentário, nome de métodos, funções e variáveis inadequados e problemas de indentação (Barnes, 2010).

Programas escritos por pesquisadores podem ser desde pequenos *scripts*, desenvolvidos para plotar gráficos e tabelas, realizar processamento matemático e estatístico e dar suporte a diversos tipos de simulações, até grandes rotinas para processar e filtrar dados das formas mais complexas. Também existem situações em que uma coleção de soluções computacionais simplesmente é “colada” para formar um único programa. Com frequência, essas soluções são desenvolvidas de uma forma tão obscura que após a publicação dos resultados, eles são simplesmente esquecidos ou deletados, fato esse que justifica a não divulgação por parte dos pesquisadores do código fonte dos seus softwares (Barnes, 2010). Mesmo quando se consideram pesquisadores que em tese possuem familiaridade com ferramentas e técnicas de software, a situação se mantém. Segundo uma pesquisa realizada com a comunidade

de aprendizagem de máquina, cerca de 78% dos entrevistados afirmaram que possuem dificuldades para compartilhar seu código fonte, fato esse causado pelo tempo dispendido com preparação, limpeza e documentação dos mesmos (LeVeque et al., 2012). Erros em softwares são bastante comuns (valores nulos, erros de conversão de dados), mas que sua detecção e correção são prejudicadas pelo não acesso ao código fonte (Morin et al., 2012).

Contudo, isso representa apenas a ponta do “iceberg”. Existem mais dificuldades em torno da pesquisa computacional quantitativa, por exemplo, os algoritmos encontrados nos artigos estão escritos em linguagem natural (português, inglês, dentre outras), o que causa problemas de ambiguidade, dificultando a sua implementação por outros pesquisadores (Ince et al., 2012). As publicações científicas baseadas em computação são atualmente fonte de erros, visto que não existem mecanismos eficientes para identificá-los e retirá-los do trabalho computacional (Donoho, 2010).

Erros são comuns na computação científica, sendo necessário trabalhar de forma árdua e enérgica para eliminá-los. Uma ferramenta para auxiliar a identificar e corrigir os erros é a replicação dos resultados computacionais obtidos mas, segundo Donoho (2010), tais resultados são apresentados de forma pouco convincente em periódicos, conferências e livros, devido principalmente as limitações impostas pela formato dessas mídias. Como consequência direta disso, observam-se que não são fornecidos subsídios suficientes para a replicação dos resultados computacionais obtidos a partir desses artigos. Para consistir resultados científicos é necessário replicar os resultados (Eckel & Peng, 2009).

A replicação é o ato de tratar um conjunto independente de dados de maneira similar à pesquisa original e então usá-la para lidar com questões científicas similares, considerada um fato necessário para atingir um alto padrão de verificação, pois outros pesquisadores podem referendar ou refutar hipóteses a partir da mesma amostra de dados da pesquisa original (Eckel & Peng, 2009).

A replicação ou reprodutibilidade já está incorporada ao cotidiano de trabalho de pesquisadores de diversas áreas. Em disciplinas teóricas, tal como a matemática, temos que uma prova formal define passos para que qualquer pesquisador, com o conhecimento necessário, reproduza a verificação de, por exemplo, um teorema. Em ciências experimentais, tais como biologia, física ou química, os experimentos devem ser replicados para que os resultados sejam validados (Fadili et al., 2010).

Entretanto, tal fato não ocorre com a pesquisa computacional quantitativa, trazendo assim várias consequências negativas além das expostas acima. A título de exemplo são apresentados diversos problemas que podem ocorrer durante o ciclo de vida de um projeto de pesquisa computacional quantitativa, os quais, se não combatidos com práticas que possi-

bilibem reprodutibilidade dos resultados, podem comprometer ou inviabilizar o projeto:

- Na última rodada de revisão, solicitam para gerar mais resultados de simulação, e por motivo de saúde o principal pesquisador está de licença médica.
- O artigo foi publicado, e nos pedem o código para refazer as experiências. Com quem está?
- Após meses de trabalho, descobrimos um erro na modelagem e precisamos de uma versão anterior do código.
- Uma parte essencial do trabalho é de responsabilidade de um pesquisador que recebeu uma boa proposta, e assim teve que abandonar imediatamente o projeto.

A partir dos itens acima, constatamos que a manutenção das versões anteriores dos artefatos da pesquisa, bem como a facilidade de reproduzir as diversas etapas, tornam-se fatores de grande importância para o sucesso de um projeto de pesquisa computacional quantitativa.

Todavia, o termo “reprodutibilidade” pode soar como trabalho extra para muitos pesquisadores, pois impõe desafios que não fazem parte da rotina de trabalho (Donoho, 2010). Como forma de contornar esse entrave, recomenda-se o desenvolvimento de um ambiente computacional que auxilie o pesquisador a aplicar as práticas de RR em um projeto de pesquisa computacional quantitativa.

1.3 Solução proposta

Neste trabalho é apresentado o desenvolvimento de um ambiente computacional para suporte as práticas recomendadas por *Reproducible Research*. Tal ambiente deve garantir que as premissas de RR sejam aplicadas de forma transparente, e que a sua adoção cause o mínimo de impacto no modo de trabalho do pesquisador. A partir do princípio de que a ciência é universal, ou seja, deve ser acessível de forma livre a todos, aplica-se esse mesmo objetivo na concepção do ambiente computacional. Para tanto, torna-se primordial a utilização de ferramentas FLOSS na sua construção. Essas ferramentas são importantes no contexto do ambiente proposto, pois estão sob licenças *General Public License (GPL)*², *Berkeley Software Distribution (BSD)*³ ou similares, empregam formatos de arquivos não proprietários, são, de maneira geral, softwares estáveis, consolidados e em constante atualização,

²Maiores detalhes em <http://www.gnu.org/licenses/gpl.html>

³Maiores detalhes em <http://opensource.org/licenses/BSD-3-Clause>

além de possuírem uma imensa comunidade de usuários e desenvolvedores que trabalham em prol das suas melhorias.

Segundo a *Free Software Foundation* (FSF)⁴, os softwares sob essas licenças garantem ao usuário: (i) processar o programa para qualquer finalidade; (ii) estudar o funcionamento do programa e modificá-lo para atender a quaisquer outras finalidades; (iii) distribuir cópias do programa a terceiros; (iv) aprimorar o programa e disponibilizar cópias das versões modificadas ao público.

Apoiando-se nas garantias acima, o ambiente computacional será desenvolvido baseado na integração de ferramentas FLOSS que foram selecionadas a partir do mapeamento das práticas de RR com funcionalidades oferecidas por tais softwares. Como vantagem dessa estratégia, além das expostas acima, temos a redução do custo de implementação e a possibilidade de distribuição de cópias gratuitas do ambiente.

1.4 Objetivos

Os principais objetivos relacionados ao trabalho são:

- Modelagem do ambiente computacional segundo a solução proposta na Seção 3.2;
- Desenvolvimento do ambiente computacional proposto;
- Disponibilizar o ambiente computacional de forma gratuita;
- Implantar o ambiente no Laboratório de Computação Científica e Visualização e no Laboratório de Computação Científica e Análise Numérica da Universidade Federal de Alagoas.

1.5 Metodologia

O ambiente computacional proposto é desenvolvido a partir da interseção do resultado de três frentes de trabalho: estudo das principais práticas de *Reproducible Research*, análise das soluções computacionais encontradas na literatura e levantamento e integração de softwares FLOSS, de acordo com as seguintes etapas:

⁴Para maiores detalhes acesse www.fsf.org

- Levantamento dos conceitos de *Reproducible Research*, através da investigação do estado da arte;
- Identificação das principais práticas recomendadas por RR;
- Análise de soluções computacionais de suporte a RR encontradas na literatura;
- Análise de ferramentas FLOSS que possam auxiliar no cumprimento de práticas RR;
- Integração de ferramentas FLOSS com o intuito de construir um ambiente computacional.

1.6 Contribuições

Disponibiliza-se um ambiente computacional especialmente desenvolvido para auxiliar laboratórios, grupos e pesquisadores interessados em realizar pesquisa computacional quantitativa comprometida com as práticas de *Reproducible Research*. Esse ambiente foi concebido para que, uma vez que esteja plenamente operacional, ofereça uma solução mais abrangente em relação aos trabalhos apresentados na literatura, com diversos benefícios associados, entre os quais destacam-se:

- Possibilitar a reprodução de estágios anteriores da pesquisa;
- Permitir que outras pessoas reproduzam etapas da pesquisa;
- Facilitar a integração de novos componentes em uma equipe de um projeto de pesquisa computacional quantitativa;
- Indiretamente, facilitar a comparação de técnicas e métodos propostos com outros disponíveis na literatura;
- Facilitar a auditoria sobre os investimentos na pesquisa;
- Prover transparência para os projetos de pesquisa.

O capítulo apresentou a motivação, solução proposta, objetivos, metodologia e contribuições inerentes ao trabalho. No próximo capítulo serão apresentados conceitos e definições, práticas de RR, vantagens e desvantagens, além da análise de ferramentas computacionais que visam auxiliar na composição da proposta do ambiente computacional.

2

Conceitos e iniciativas de *Reproducible Research* na ciência computacional

A reprodutibilidade dos resultados é um fator primordial na ciência. A credibilidade e o sucesso de uma pesquisa estão ancoradas na vontade dos cientistas de expor suas ideias e resultados a testes independentes e replicação por outros cientistas. Isso requer uma completa abertura de dados, procedimentos e materiais (Fomel & Claerbout, 2009).

2.1 Práticas de *Reproducible Research*

O progresso da pesquisa computacional quantitativa inevitavelmente deve ser avaliado pela reprodução dos resultados publicados, embora tal avaliação seja dificultada por diversos fatores mencionados em seções anteriores deste trabalho. Mas, com a missão de solucionar esse problema, um grupo de pesquisadores, editores e representantes de agências de fomento formularam um conjunto de passos necessários para facilitar a reprodução da pesquisa (Stodden et al., 2010). Como resultado final desse trabalho estão as propostas a seguir.

Os pesquisadores devem colocar seus códigos e dados:

- Em uma página web (instituição ou laboratório);
- Em um projeto aberto de compartilhamento de código fonte, como *sourceforge*, *bitbucket* ou *github*, entre outros disponíveis.

Como recomendações para os pesquisadores, temos:

- Garantir acesso irrestrito a todas as alterações de dados e código fonte. Isso pode ser facilmente resolvido com a utilização de ferramentas de controle de versão;
- Incluir descrições detalhadas do ambiente utilizado, da versão de software e de links estáveis contendo dados e códigos;
- Licenciar o código produzido como aberto (GPL ou BSD preferencialmente), facilitando o reuso como sugerido pelos padrões de *Reproducible Research*;
- Usar um acesso aberto de publicação de artigos, e avaliar os rascunhos em sites como *arXiv*¹ e *PubMed Central*², dentre outros;
- Encorajar reuso e uso de padrões, publicando dados e códigos em formatos não proprietários.

As recomendações para as agências de fomento são:

- Estabelecer um serviço de hospedagem de código e dados, com possibilidade de acesso a todas as alterações realizadas nos arquivos submetidos;
- Fomentar e encorajar o desenvolvimento de um conjunto de práticas para a pesquisa;
- Criar um conjunto de ferramentas que integre código e dados, incluindo técnicas de padronização das publicações;
- Criar ferramentas de fluxo para o compartilhamento de dados e arquivos, entre diferentes instituições.

Para os editores e periódicos as recomendações são:

- Implementar políticas de auditoria dos dados e códigos publicados;
- Se for possível, replicar os resultados computacionais da publicação.

Com base nas recomendações definidas acima, foram elaboradas as práticas de *Reproducible Research*, que devem ser atendidas por qualquer pesquisa em ciência computacional quantitativa. Em suma, tais práticas recomendam que toda pesquisa, para ser aceita em veículos de impacto (periódicos e conferências em nível internacional), possua as seguintes características e funcionalidades:

¹<http://arxiv.org/>

²<http://www.ncbi.nlm.nih.gov/pmc/>

- Esteja associada a um repositório com acesso público pela web;
- Sejam disponibilizados o relatório da pesquisa, o(s) conjunto(s) de dados, *scripts*, comandos, instruções e qualquer artefato relevante;
- Esteja associada a um fórum ou algo similar que possua recursos de discussão entre autores e leitores;
- Que seu relatório possua o formato de arquivo IMRAD (Introdução, Metodologia, Resultados e Discussão).

Além disso, é necessário que a pesquisa computacional quantitativa esteja associada obrigatoriamente a uma página web de acesso público que contenha as seguintes informações:

- Título da pesquisa;
- Autores, com links às respectivas páginas web, por exemplo, link para o currículo Lattes;
- Resumo da pesquisa;
- O artigo em PDF, e informações sobre seu estado (em preparação, submetido, aceito);
- Os códigos que geram os dados, as figuras e as tabelas, documentado e com informações sobre como executá-lo;
- Os dados empregados, e informações sobre como lê-los;
- Informações da plataforma onde o software foi testado;
- Email para contato;
- Referências com os seus *abstracts*.

2.2 Vantagens e desvantagens da utilização de *Reproducible Research*

Reproducible Research é uma sistematização essencial para demonstrar que os resultados apresentados em um artigo científico foram alcançados. Facilita a reprodução dos resultados por outros pesquisadores, e por consequência, auxilia a auditoria das agências de fomento.

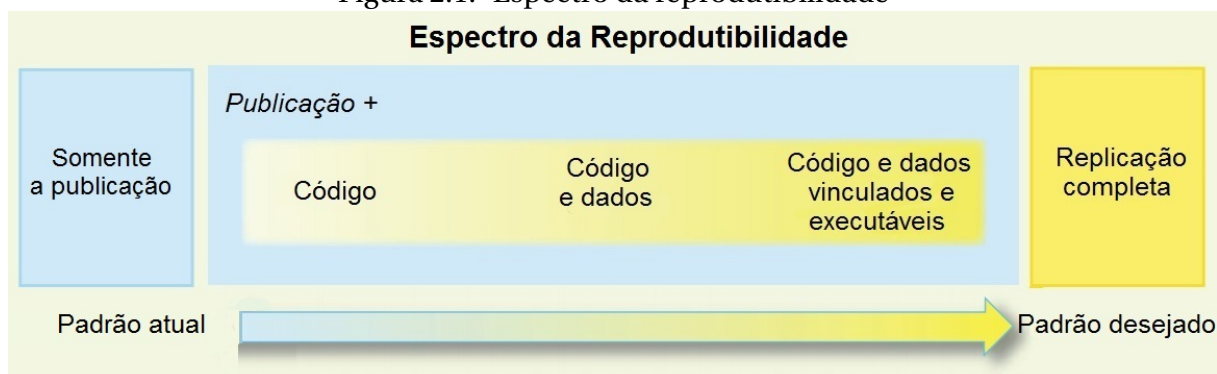
A desvantagem pode ser mensurada pelos impactos causados ao projeto de pesquisa, na medida que, para implantar práticas de *reproducible research*, uma carga extra de trabalho é exigida para todos os integrantes. Essa carga é causada pelo esforço em atender de forma satisfatória as demandas de RR, tais como: construção e manutenção de um sítio web, versionamento de todos os artefatos relevantes da pesquisa e a geração de manuais, descritores e documentação. Tais demandas também acarretam mudanças no modo de trabalho do pesquisador, pois, com frequência, essas atividades não fazem parte da sua rotina.

2.3 *Reproducible Research* na ciência computacional quantitativa

Diversas conferências e periódicos de prestígio internacional têm atribuído mais relevância a *Reproducible Research*, tanto que suas políticas para publicação de artigos estão sendo revistas com a intenção de exigir cada vez mais a disponibilização de artefatos que auxiliem a reprodução dos resultados. O periódico “*Science*” incluiu o código fonte do artigo na lista de itens a ser submetido pelo autor. O periódico “*Biostatistics*” nomeou um editor responsável pela reprodutibilidade dos resultados dos artigos submetidos (Ince et al., 2012). Também vale destacar a crescente adoção das práticas de RR nos periódicos “*International Journal of Forecasting*”, “*Biometrical Journal*”, “*IEEE Transactions on Signal Processing*” na conferência “*International Conference on Image Processing*” e as iniciativas www.ipol.im, <http://www.openresearchcomputation.com/>, <http://www.rrplanet.com/> e <http://icy.bioimageanalysis.org/>.

Como forma de facilitar o atendimento das novas políticas para publicação de artigos, os pesquisadores têm recorrido a soluções computacionais que vão desde *scripts* para automatização das diversas etapas da confecção de um artigo até ferramentas que disponibilizam armazenamento de artefatos em um repositório público na internet. Para ilustrar as vantagens da utilização de ferramentas computacionais apresenta-se a Figura 2.1, adaptada de Peng (2011). Tal figura mostra o árduo “caminho” para tornar uma publicação reprodutível.

Figura 2.1: Espectro da reprodutibilidade



Fonte: Imagem adaptada de Peng, 2011, p. 3.

O estudo de tais soluções e ferramentas têm importância para a proposta do ambiente computacional, já que proporciona uma análise de funcionalidades, estrutura e interface visual que podem ser aproveitados no ambiente.

2.3.1 Análise de soluções computacionais

Para esta análise, foram selecionadas ferramentas que auxiliem a reprodução de resultados computacionais, bem como ferramentas que sincronizem código fonte e relatórios de pesquisa. O estudo de tais soluções possui grande importância na medida que, possuem funcionalidades que podem ser incorporadas na proposta do ambiente computacional. Além das ferramentas analisadas nas seções abaixo, destacam-se as soluções computacionais encontradas em Eaton (2012) e Li-Thiao-Te (2012).

WaveLab e MCALab

WaveLab é uma biblioteca de rotinas em *MATLAB* que podem ser utilizadas para implementar uma série de algoritmos relacionados à análise *wavelet*. *WaveLab* torna disponível, em um único pacote, todo o código necessário para reproduzir os resultados de forma mais independente possível do desenvolvedor. O pesquisador interessado pode inspecionar o código fonte verificando exatamente como os algoritmos e parâmetros são usados, podendo inclusive modificá-los para produzir seus próprios resultados (Buckheit & Donoho, 1995). Para maiores detalhes consulte <http://www-stat.stanford.edu/~wavelab/>.

MCALab também é uma biblioteca de rotinas em *MATLAB* que descreve e implementa uma série de algoritmos relacionados à decomposição de sinais e imagens e ao processo de

inpainting. O software possui dois diretórios principais, um para sinais (1D) e outro para imagens (2D). Cada diretório possui a seguinte estrutura de sub-diretórios: *decomposition*, *dictionaries*, *datasets*, *demoGui* e *scripts* (Fadili et al., 2010). Uma estrutura de diretórios análoga é utilizada no ambiente computacional.

As duas soluções são bastantes restritivas, pois tratam-se de rotinas desenvolvidas com o software proprietário *MATLAB* e a área de atuação das duas limita-se a análise de *wavelet*, decomposição de imagens e *inpainting*. Esses fatos impedem que os softwares atinjam um público maior, além de tornar mais árdua a tarefa de reprodução dos resultados, pois exigirá que os pesquisadores possuam licenças do *MATLAB*.

ReDoc

O *ReDoc* é um sistema para reprodução da computação científica em documentos eletrônicos que foi implementado no início dos anos 90 nas dependências da Universidade de *Stanford*. O software foi concebido a partir da necessidade de diminuir o tempo e esforço despendidos na reprodução de resultados computacionais.

ReDoc é um software de armazenamento de artefatos que auxilia o trabalho colaborativo em pesquisa. Pesquisadores podem publicar seus resultados e permitir que outros o reproduzam, incrementando melhorias e inovações à pesquisa original. Para que seja possível a reprodução dos resultados é necessário que os pesquisadores utilizem um conjunto de regras e comandos específicos do software (Schwab et al., 2000).

Tal como as ferramentas *WaveLab* e *MCALab*, a solução mostra-se restritiva apenas às áreas da ciência que são alvo de pesquisas do laboratório onde a ferramenta foi desenvolvida.

Image Processing On Line (IPOL)

O *IPOL* é um periódico com foco em pesquisas na área de processamento e análise de imagens. Cada artigo publicado deve disponibilizar obrigatoriamente a descrição detalhada do algoritmo utilizado, a sua implementação nas linguagens de programação *C* ou *C++* e a demonstração on-line (disponível no endereço www.ipol.im) onde o algoritmo pode ser testado com conjunto de dados carregados pelos usuários. Um dos objetivos do *IPOL* é tornar acessíveis os algoritmos contidos nos artigos publicados.

A solução apresentada pelo *IPOL* auxilia sobremaneira a reprodução dos resultados dos artigos publicados, já que torna público (via internet) todos os artefatos da pesquisa, obri-

gando os autores a descrever detalhadamente os algoritmos, os dados utilizados e os resultados obtidos. Uma página web similar é utilizada no ambiente computacional proposto, entretanto, a solução *IPOLE* restringe-se à área de processamento e análise de imagens e a algoritmos implementados nas linguagens *C* e *C++*.

Sweave

Um trabalho de análise estatística típica consiste de duas etapas, a análise de dados a partir de um software apropriado e a elaboração de um relatório que contenha as principais conclusões extraídas na análise. De forma geral, essas duas etapas acontecem paralelamente. Após diversas iterações nas duas etapas é comum haver inconsistências, o que pode comprometer a qualidade do trabalho, forçando os pesquisadores a se preocuparem com a questão da sincronização das etapas.

O *Sweave* surgiu como uma resposta ao problema de sincronização dessas etapas. Seu principal propósito é a criação de relatórios dinâmicos através da unificação da análise estatística e do relatório de análise em um só arquivo (Leisch, 2002). Para isso, o *Sweave* utiliza as linguagens de programação *R* ou *S-Plus* como ferramenta de análise estatística e o \LaTeX como plataforma de elaboração do relatório. Existem diversos benefícios associados a essa integração, tais como geração e inclusão automática de gráficos e dados da análise no relatório de pesquisa. O *Sweave* faz parte da instalação normal da linguagem *R* (versão 1.5.0 ou superior) disponível no pacote *Utils* (Leisch, 2002).

StatWeave

O *StatWeave* é um software desenvolvido com o intuito de fornecer integração entre código fonte e documentação para uma grande variedade de linguagens (*R*, *SAS*, *S-plus*, *Maple*) e formatos de arquivo (\LaTeX e *odt*). Sua concepção foi baseada nas ferramentas *Sweave* (Leisch, 2002) e *SASweave* (Lenth & Hojsgaard, 2007). *StatWeave* é implementado na linguagem *Java*, o que garante compatibilidade com diversas plataformas de sistema operacional. Maiores detalhes podem ser obtidos em <http://homepage.stat.uiowa.edu/~rlenth/StatWeave/>.

Org-mode

O *Org-mode* é uma extensão para o software de edição de documentos de texto *Emacs* que provém uma linguagem de marcação para criar, analisar e interagir com documentos de

texto hierarquicamente organizados. Possui funcionalidades que incluem estruturação de texto, gerenciamento de projeto e um sistema de publicação que possui funcionalidades que permitem gerar relatórios em uma variedade de formatos, tais como \LaTeX , *HTML* e *DocBook* (Schulte et al., 2012).

Essa extensão possui o mesmo princípio de funcionamento do *Sweave*, garantindo consistência entre código fonte e texto. De maneira geral é uma solução mais genérica, pois possui mais formatos de saída de relatório e suporte a diversas linguagens de programação. A forma encontrada pela ferramenta de garantir sincronização entre dados e texto foi definir uma estrutura rígida em seus documentos (arquivos).

As principais estruturas de um documento *Org-mode* são os códigos ativos e os códigos de texto. No código ativo estão inclusos os trechos de código fonte e dados que são avaliados no momento da geração do relatório, com o objetivo de retornar seu resultado computacional. Tais resultados podem ser expressos como gráficos, matrizes, tabelas, dentre outros. O resultado de um código ativo pode ser nomeado e referenciado por outros trechos de código de texto. O *Org-mode* permite a existência de código fonte de diversas linguagens computacionais em um único documento e, caso seja necessário, o software é suficientemente flexível para suportar novas linguagens (Schulte et al., 2012). Maiores detalhes podem ser obtidos em <http://orgmode.org/>.

As soluções *Sweave*, *StatWeave* e a *Org-mode* analisadas anteriormente, possuem a finalidade de sincronizar resultados computacionais obtidos a partir da execução de um trecho de código de fonte com um relatório de pesquisa. A diferença entre as ferramentas basicamente se concentra no suporte a linguagens de programação e no formato de saída do relatório de pesquisa. Essas soluções auxiliam o trabalho colaborativo em um projeto de pesquisa, mas não tratam das questões relacionadas às práticas de RR, tais como versionamento de artefatos, geração de página web, dentre outras.

A Tabela 2.1 mostra um quadro comparativo com as principais soluções para RR baseado nas suas funcionalidades. Para essa comparação, alguns aspectos são analisados:

- Sincronização entre código fonte e relatório;
- Atendimento as práticas de *reproducible research*;
- Genérica, ou seja, se a solução atende a qualquer área da pesquisa computacional quantitativa;
- Linguagem de programação que a solução permite.

Tabela 2.1: Quadro comparativo de soluções

Software	S. Texto/Código	Práticas RR	Genérica	Ling.
Wavelab/MClab	-	-	-	MATLAB
Redoc	-	-	-	-
Sweave	X	-	X	R e S-Plus
StatWeave	X	-	X	Todas
Org-mode	X	-	X	Todas
IPOL	-	X*	-	C e C++
Solução proposta	-	X	X	Todas

Fonte: Autor, 2013.

O capítulo apresentou conceitos e práticas de RR, vantagens e desvantagens e a análise de ferramentas computacionais que são utilizadas para atender às práticas de RR. No próximo capítulo serão apresentados todos os requisitos e a proposta do ambiente computacional.

3

Requisitos e proposta do ambiente computacional

A partir das práticas e da análise de soluções e iniciativas para RR existentes na literatura, foi possível estabelecer os principais requisitos que nortearam o ambiente computacional proposto neste trabalho.

3.1 Requisitos para um ambiente computacional para suporte a *Reproducible Research*

A partir da definição das práticas de RR, foi possível realizar um mapeamento para funcionalidades de software. Tal mapeamento permite que os conceitos de RR, que podem ser automatizados por ferramentas computacionais, sejam atendidos de forma plena. Na Tabela 3.1, é apresentado o mapeamento utilizado no ambiente computacional proposto.

Tabela 3.1: Mapeamento das principais funcionalidades do ambiente

Prática	Funcionalidade
Repositório público na web	Gerenciador de conteúdo web
Versionamento dos artefatos da pesquisa	Software para controle de versão
Recursos de discussão	Ferramenta wiki
Armazenamento de dados	Banco de dados
Geração do artigo IMRAD	Gerenciador de conteúdo Web

Fonte: Autor, 2013.

3.2 Proposta para o ambiente computacional

O ambiente computacional proposto foi desenvolvido baseado na integração de ferramentas FLOSS. Essa abordagem visa diminuir o custo do desenvolvimento, pois os softwares utilizados na composição do ambiente encontram-se em um estágio pleno de utilização, possui grande comunidade de usuários e desenvolvedores que interagem garantindo atualizações, testes e melhorias em suas funcionalidades. Assim, não é mais necessário “reinventar a roda”, implementando softwares com funcionalidades encontradas facilmente na comunidade FLOSS.

Para que a integração das ferramentas seja bem sucedida é imprescindível a construção de um núcleo central, que garanta a comunicação entre esse núcleo e os softwares FLOSS integrantes do ambiente. Outro fator importante é a possibilidade de mudança dos softwares integrantes, já que softwares podem evoluir e também serem descontinuados. Dessa forma, o ambiente não deve estar atrelado a uma certa versão de software, tem que haver possibilidade de mudança mesmo que isso exija um esforço de programação.

Um conceito relevante na proposta é o projeto de pesquisa *Reproducible Research*, o qual define em termos de organização de diretórios e arquivos, um projeto de pesquisa para a ferramenta. Para tanto, é necessário que o ambiente computacional possua as seguintes funcionalidades:

- Armazenamento do projeto utilizando um servidor com disponibilidade na internet (Servidor de Projetos RR);
- Cada projeto será fisicamente armazenado em um diretório com as permissões de escrita/leitura para cada pesquisador envolvido no projeto;
- O acesso ao projeto por parte do pesquisador será através de um navegador web *browser*;
- Backup, versionamento dos arquivos e histórico completo de alterações garantidos de forma transparente;
- O projeto deve ser composto de subdiretórios (*scripts/código*, dados, textos e descritores) de acordo com que o preconiza *Reproducible Research*, podendo ser modificado a partir da necessidade da pesquisa.

Outro conceito importante é a versão da pesquisa. Trata-se da série de resultados alcançados em um projeto de pesquisa ao longo do tempo. De forma geral, à medida que um

projeto de pesquisa evolui, os seus resultados vão se modificando, pois passam a ser comparados com novos métodos e técnicas. O ambiente computacional proposto preocupa-se em reproduzir os resultados alcançados por uma versão do projeto de pesquisa.

Para trabalhar com essas estruturas, serão criados três módulos: gerência de projeto RR, suporte a projeto RR e consulta aos projetos RR. Todos os módulos terão as suas funcionalidades destacadas abaixo. O módulo de gerência de projetos RR define aspectos básicos e estruturantes necessários para o início da utilização do ambiente computacional:

- Cadastro de pesquisadores;
- Cadastro de projetos;
- Cadastro e associação de artefatos para o projeto;
- Associação entre pesquisadores e projetos;
- Geração da versão da pesquisa;
- Criação de página web do projeto;
- Integração com o servidor de Projetos RR.

O módulo de suporte a RR, desenvolvido para garantir o provimento das características de *Reproducible Research* aos projetos, possui as seguintes funcionalidades e características listadas abaixo:

- Disponibilizado em ambiente web;
- Desenvolvido para permitir suporte completo a projetos RR;
- Disponibiliza cópia dos artefatos da pesquisa na estação de trabalho do pesquisador;
- Permite a utilização das ferramentas usuais dos pesquisadores.

Um aspecto importante no ambiente é relacionado com o modo de trabalho do pesquisador, o qual não deve sofrer qualquer alteração significativa. Esse detalhe torna-se importante no sentido de popularizar o ambiente, contribuindo fortemente para a sua adoção.

O módulo de consulta a projetos RR permite que outros pesquisadores consultem versões de projetos disponibilizadas ao público e possui as seguintes características e funcionalidades listadas abaixo:

- Disponibilizado em ambiente web;
- Consulta por título, pesquisador, palavras-chave e outros;
- Disponibiliza ferramenta de comunicação com os autores;
- Permite o *download* de todos os artefatos de uma versão de um projeto de pesquisa.

3.3 Requisitos adicionais para o ambiente computacional

Requisitos adicionais que possuem importância destacada para o ambiente computacional proposto são: os direitos autorais dos artefatos desenvolvidos durante o ciclo de vida de uma pesquisa e questões relacionadas a segurança e certificação dos artefatos.

O conceito básico de direito autoral define que os trabalhos publicados e não publicados em diversas áreas, tais como: literatura, música, fotografia, coreografia, escultura e outros trabalhos visuais de arte, como programas de computador, são protegidos. O direito autoral preserva a expressão de ideias e reserva para seus autores o direito exclusivo de reproduzir seus trabalhos, podendo o mesmo permitir através de autorização prévia e expressa, a reprodução parcial ou total por outros.

Verifica-se que o direito autoral não se adequa às normas científicas para a reprodução dos resultados de pesquisa, pois elas defendem que os resultados sejam reproduzidos por qualquer pesquisador sem que haja necessidade de autorização expressa. Diante disso, torna-se necessária a utilização de mecanismos que assegurem a total reprodutibilidade da pesquisa por parte de outros, bem como proteja os direitos do autor da pesquisa original, garantindo por exemplo, a citação (Stodden, 2009).

Esse mecanismo deve prover suporte a artefatos de código (*scripts*, documentação de código entre outros), mídias (dados brutos, imagens, tabelas, figuras, entre outros) e acesso aos dados em banco de dados, que se constituem como os principais artefatos de uma pesquisa computacional quantitativa. Para a resolução desse problema, apresenta-se o *Reproducible Research Standard* (RRS) (Stodden, 2009). O RRS sugere a adoção de licenças públicas consolidadas mundialmente para os diferentes aspectos dos artefatos da pesquisa. Dessa maneira, artefatos de mídias devem seguir a licença *Creative Commons CC BY*¹. Todo código fonte deve aderir à licença BSD, e dados contidos em bases devem atender ao protocolo *open-access-data-protocol*².

¹Maiores detalhes em <http://creativecommons.org/licenses/>

²Maiores detalhes em http://wiki.creativecommons.org/Open_Access_Data_protocol

Questões de segurança e certificação estão relacionadas à adoção de recursos que garantam autenticidade, confidencialidade e integridade às informações eletrônicas. Para esse propósito é necessário que o ambiente computacional suporte elementos que permitam a assinatura digital de qualquer artefato de texto. Em suma, a assinatura digital possui um método de autenticação de informação digital, em geral criptográfico, sendo considerado análogo à validade da assinatura física em papel. A utilização da assinatura digital assume a prova irrefutável de que a mensagem ou documento veio do emissor.

3.4 Softwares FLOSS componentes do ambiente

São softwares licenciados de forma a conceder aos seus usuários os direitos de utilização, estudo e modificação, disponibilizando seu código fonte de forma livre. Esta abordagem ganhou impulso a partir da aceitação dos potenciais benefícios exercidos por essa forma de licença.

Segundo a *Open Source Initiative* (OSI)³ um software com licença *Open Source* segue uma lista de requisitos, dos quais destacamos: (i) garantia da liberdade de redistribuição do software; (ii) disponibilização do código-fonte do software; (iii) liberdade de modificação do software e de elaboração de softwares derivados, permitindo-se que sejam redistribuídos sob os mesmos termos de licença do software original; (iv) vedação de discriminação de quaisquer pessoas ou grupos; (v) vedação de discriminação do campo de atuação onde se utilizará o software.

Softwares com esses requisitos são importantes neste trabalho, na medida em que podem ser modificados, integrados, distribuídos e utilizados sob qualquer propósito. Dessa forma, uma atividade essencial é a análise de ferramentas FLOSS, sempre com o intuito de verificar se as funcionalidades oferecidas por elas atendem à alguma prática de RR. Nesse caso, os softwares escolhidos para a fase de análise devem possuir funcionalidades relativas a:

- Gestão de página web (*Joomla, Plone, Wordpress*);
- Gestão de versões de artefatos (*SVN, GIT, Mercurial*);
- Gestão de fóruns (*MediaWiki*);
- Gestão de armazenamento de dados (*MySQL, PostgreSQL*);

³Para maiores detalhes consulte <http://opensource.org/>

Os softwares escolhidos são encontrados na seção 3.4.5.

3.4.1 Gestão de página web

Uma solução amplamente utilizada para geração e atualização rápida e eficiente de páginas web são os sistemas de gerenciamento de conteúdo (*Content Management System* (CMS)). Dentre as funcionalidades oferecidas por esses sistemas, estão: criar e gerenciar um conjunto de páginas web, atualização rápida de conteúdo, facilidade de utilização por parte dos usuários. Tais softwares permitem separação entre design, programação e conteúdo, o que garante flexibilidade na produção de sites com interface customizada.

O software escolhido será utilizado para a gestão da página web do projeto de pesquisa e para a geração do relatório no formato IMRAD.

Joomla

Joomla é um sistema de gestão de conteúdo bastante utilizado para o desenvolvimento de lojas virtuais, blogs, revistas, portais, catálogos online, dentre outros formatos de conteúdo na web. Possui milhares de extensões que permitem agregar novos serviços e funcionalidades de maneira rápida e eficiente. É desenvolvido na linguagem de programação *PHP*, utiliza banco de dados *MySQL* e pode ser executado sob as plataformas *Windows* e distribuições *Linux*. Maiores detalhes podem ser obtidos em <http://www.joomla.org/>.

Plone

Plone é um sistema de gestão de conteúdo desenvolvido na linguagem *Python*, que executa sobre um servidor de aplicações chamado *Zope* e sobre o framework *CMF Content Management Framework*. O *Plone* é bastante utilizado para a construção de portais de informação que suportem grande quantidade de acessos e que sejam executados em ambientes de *intranets*, *extranets* e na internet. Além disso, diversos usuários têm usado-o para desenvolver sistemas de publicação de documentos ou até como ferramenta para trabalho colaborativo. Maiores detalhes podem ser obtidos em <http://plone.org/>.

WordPress

WordPress é um dos softwares CMS mais utilizados para a criação de blogs. É amplamente adotado por usuários que exijam blogs profissionais e com recursos diferenciados. O *WordPress* é utilizado também como plataforma de desenvolvimento de sites de comércio eletrônico, revistas, jornais, portfólios, gerenciador de projeto, diretório de eventos e outros conteúdos devido a sua capacidade de extensão através de plugins e temas customizáveis. Maiores detalhes podem ser obtidos em <http://wordpress.org/>.

3.4.2 Gestão de versões de artefatos

Um software para controle de versão tem como principal finalidade o gerenciamento das diversas versões e histórico de alterações no desenvolvimento de um artefato qualquer (texto, imagens, dados, código fonte, dentre outros). No caso específico do ambiente proposto, essa funcionalidade atende à prática de RR que enfatiza a recuperação de versões anteriores de artefatos e fases da pesquisa.

O software escolhido será utilizado para o armazenamento de todas as versões dos artefatos do projeto de pesquisa.

SVN

O *SVN* é um projeto *open source* iniciado em meados dos anos 2000. Possui como principal funcionalidade o suporte ao versionamento de arquivos. A partir de um arquivo versionado é possível usar as ações a seguir: (i) voltar a versões anteriores do arquivo. (ii) verificar o histórico de mudanças. (iii) verificar qual usuário modificou o arquivo. O *SVN* é uma ferramenta bastante popular com uma grande comunidade de usuários que disponibiliza documentação, fóruns e ajuda. Maiores detalhes podem ser obtidos em <http://subversion.tigris.org/>.

GIT

O *GIT* é um sistema de controle de versão com características distribuídas, ou seja, cada diretório de trabalho representa um repositório que contém todo o histórico de alterações dos arquivos versionados. Tal característica proporciona ao sistema sua utilização sem a necessidade de acesso a um servidor central. O *GIT* possui suporte para desenvolvimento

não-linear, o que leva à criação de muitos ramos comumente conhecidos como *branches* e mescla conhecidos como *merge* com performance superior em relação aos seus concorrentes para grandes projetos. Essas funcionalidades foram implementadas a partir das demandas geradas por um projeto altamente distribuído, por exemplo, desenvolvimento de uma nova distribuição para o sistema operacional *Linux*, onde milhares de desenvolvedores mundo afora contribuem para o projeto. Maiores detalhes podem ser obtidos em <http://git-scm.com/>.

Mercurial

Mercurial é uma ferramenta multi-plataforma de controle de versão distribuída com utilização apropriada para projetos de desenvolvimento de softwares. Maiores detalhes podem ser obtidos em <http://mercurial.selenic.com/>.

3.4.3 Gestão de fóruns

Categoria de software que garante formas de comunicação entre autores e leitores, através de ferramentas de email, formulários e fóruns.

MediaWiki

MediaWiki é um software *Wiki* desenvolvido na linguagem *PHP*, utilizando sistemas de gestão de base de dados *MySQL*, e licenciado sob a licença GNU GPL. Essa tecnologia é utilizada principalmente nos projetos da *Wikimedia*.

3.4.4 Gestão de armazenamento de dados

Conjunto de softwares cujo objetivo principal é gerenciar o acesso e a correta manutenção dos dados armazenados em suas bases. O software escolhido será utilizado para o armazenamento de todos os dados do ambiente computacional.

MySQL

Sistema gerenciador de bases de dados relacional, bastante popular, com milhões de instalações no mundo, possuindo como principais características a portabilidade, perfor-

mance, facilidade no uso, e estabilidade. Maiores detalhes podem ser obtidos em www.mysql.com/

PostgreSQL

Sistema gerenciador de bases de dados relacional similar ao *MySQL*. Maiores detalhes podem ser obtidos em www.postgresql.com.

3.4.5 Softwares selecionados

Devido a grande disponibilidade de softwares FLOSS existentes, foi necessário realizar uma seleção baseada nos seguintes aspectos: integração com a linguagem de programação *Java*, maturidade, ou seja, tempo em que a ferramenta está disponível gratuitamente, existência de documentação gratuita, fóruns de discussão e regularidade em atualizações de versão.

Mesmo com a utilização dos critérios de seleção, não há garantias de que os softwares escolhidos sejam as melhores opções para a composição do ambiente computacional. Entende-se que esse fato, apesar de ser considerado algo indesejado, não compromete o desenvolvimento do ambiente.

Na Tabela 3.2 são mostrados os softwares escolhidos para compor o ambiente computacional.

Tabela 3.2: Softwares escolhidos para o ambiente computacional

Prática RR/Funcionalidade	Software
Geração de página web	<i>WordPress</i>
Controle de versão	<i>SVN</i>
Recursos de discussão	<i>MediaWiki</i>
Armazenamento de artefatos	<i>MySQL ou postgresQL</i>
Geração do artigo IMRAD	<i>WordPress</i>

Fonte: Autor, 2013.

O capítulo apresentou os requisitos necessários para a construção do ambiente computacional. No próximo capítulo serão apresentados os modelos e diagramas gerados durante a fase de modelagem do ambiente.

4

Modelagem do ambiente computacional proposto

Neste capítulo são apresentados os aspectos técnicos necessários (tecnologias, ferramentas, modelagem e diagramas) que delinearão o desenvolvimento do ambiente.

4.1 Tecnologias e ferramentas utilizadas

Na implementação do ambiente computacional foi necessário utilizar as tecnologias e ferramentas descritas a seguir.

4.1.1 Linguagem de programação *Java*

Linguagem de programação escolhida para o desenvolvimento de todos os módulos do ambiente computacional. Suas características desejadas para o trabalho são: (i) suporte a orientação a objetos, (ii) portabilidade, ou seja, suas aplicações executam da mesma forma sob qualquer sistema operacional, (iii) extensa biblioteca, com rotinas para interagir com protocolos de rede e internet e (iv) suporte à diversas línguas (inglês, português, espanhol, dentre outras).

Java é bastante popular entre a comunidade de programadores, suas aplicações são encontradas nos mais variados ambientes, desde dispositivos móveis, até grandes computadores e *clusters*. Maiores detalhes em <http://www.oracle.com/technetwork/java/index.html>.

4.1.2 Mapeamento objeto-relacional *Hibernate*

Trata-se de um *Framework*¹ que possui como principal objetivo a construção de uma camada de abstração entre as classes da linguagem *Java*, representadas por objetos e as tabelas de um banco de dados relacional, representada por um conjunto de relações.

Hibernate provê um mapeamento que transforma as classes *Java* em tabelas, e vice-versa, de forma transparente. Essa funcionalidade facilita a portabilidade do banco de dados relacional no ambiente proposto. Maiores detalhes em <http://www.hibernate.org/>.

4.1.3 Interface web para o usuário *Java Server Faces*

O *Java Server Faces* é um *Framework* para auxiliar o desenvolvimento de interfaces de usuário baseadas na web. Possui como principais características: um conjunto de componentes pré-definidos e personalizados, utilização de folhas de estilo (*Cascading Style Sheets* (CSS)), suporte a *javascript* e *ajax*, internacionalização, acessibilidade, gerência de sessão de objetos e componentes, manipulação de eventos no lado cliente e no lado servidor.

No contexto do trabalho, a utilização de *Java Server Faces* como solução para o desenvolvimento da interface web acarreta como principal vantagem a adoção de uma interface visual padrão, através do uso de componentes pré-definidos e folhas de estilo. Maiores detalhes em <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>.

4.1.4 Ambiente de desenvolvimento integrado *NetBeans*

O *NetBeans* é um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software. Portável, pode ser executado em diversas plataformas de sistemas operacionais. Extensível, possui *plug-ins* para as mais diversas funcionalidades. Para o trabalho, trata-se de um ambiente único, onde todas as ações relacionadas à implementação estão concentradas, desde a criação das classes, até os testes do ambiente. Maiores detalhes podem ser obtidos em <https://netbeans.org/>.

¹“Conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um sub-sistema da aplicação.”

4.1.5 Servidor web *Apache Tomcat*

O *Apache Tomcat* é um software de código fonte aberto, que implementa as tecnologias de *Java Servlet* e de *JavaServer Pages*. Amplamente utilizado para softwares que executam na web, suas aplicações englobam desde pequenos sites caseiros, até sistemas de grande escala e de missão crítica para uma gama de organizações. Ferramenta responsável pela execução do ambiente computacional na web. Maiores detalhes podem ser obtidos em <http://tomcat.apache.org/>.

4.1.6 Modelagem UML *Jude*

O *Jude* é uma ferramenta de geração de diagramas de softwares para o padrão *Unified Modeling Language* (UML), utilizada para a geração de todos os diagramas e modelos UML do ambiente computacional. Maiores detalhes em <http://jude.change-vision.com/jude-web/index.html>.

4.1.7 Modelagem de banco de dados relacional *DIA*

O *DIA* é uma ferramenta que possui capacidade para desenhar diagramas e modelos para diversas necessidades, tais como fluxogramas, diagramas de rede, diagramas de circuitos, modelo entidade-relacionamento, dentre vários outros. Utilizada para a geração dos modelos conceitual e lógico do ambiente computacional. Maiores detalhes em <http://dia-installer.de/>.

4.2 Modelagem do ambiente computacional

Nesta seção são apresentados os requisitos funcionais e os modelos de dados do ambiente computacional.

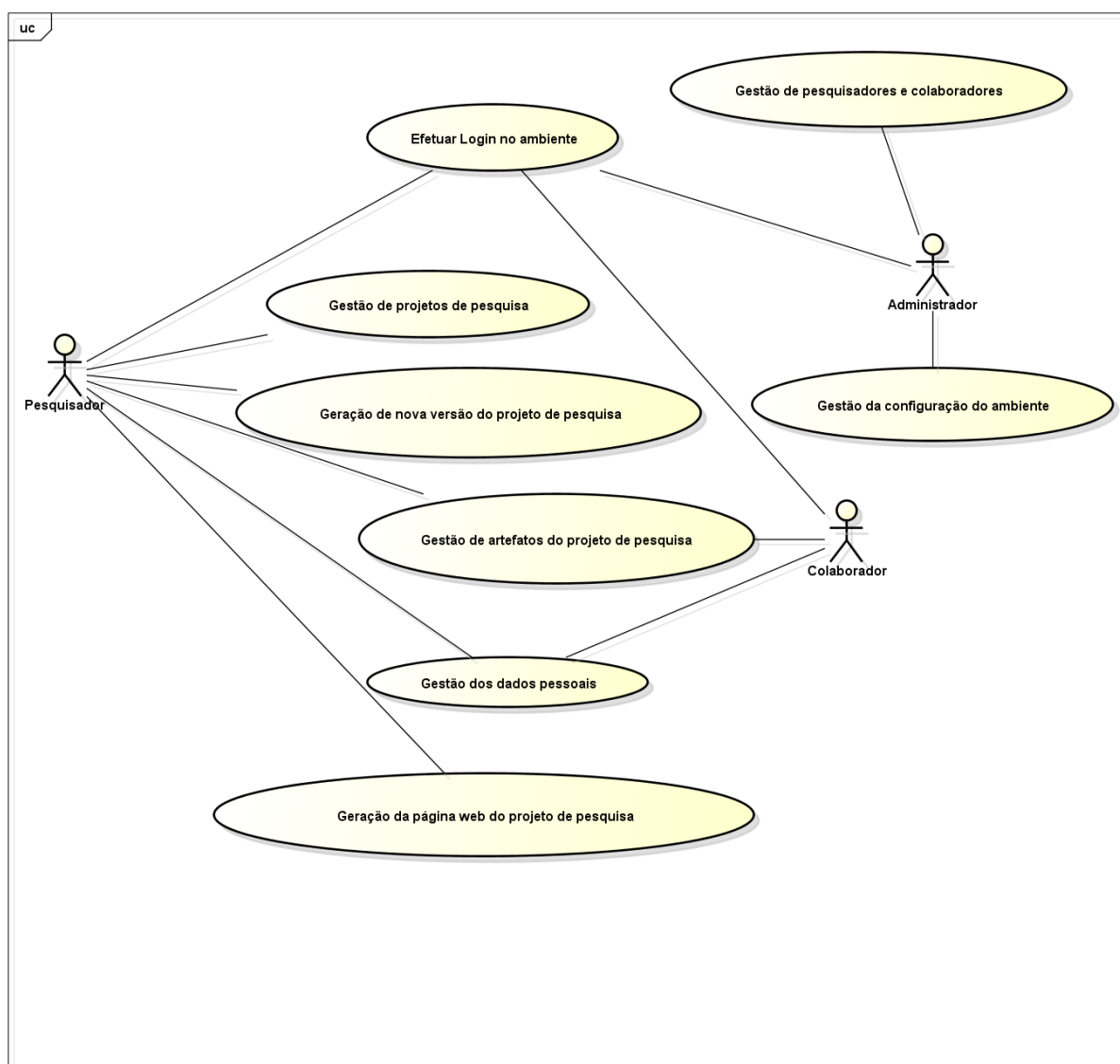
4.2.1 Requisitos funcionais

Os requisitos funcionais foram definidos nas seções 3.1, 3.2 e 3.3. Para representá-los em uma forma visual, foi escolhido o diagrama de casos de uso da UML. Os casos de uso são utilizados como representações dos requisitos funcionais de um software. Mostram funções

completas de um sistema e realizam um aspecto maior da funcionalidade. Todo caso de uso deve gerar um ou mais benefícios para o cliente ou os usuários, visíveis nas pós-condições resultantes. O conjunto completo dos casos de uso deve descrever a funcionalidade completa do sistema, sem lacunas e sem superposição; daí dizer-se que cada caso de uso representa uma “fatia de funcionalidade” (Paula Filho, 2010).

A figura 4.1 apresenta os atores² do ambiente computacional (Pesquisador, Colaborador e Administrador), com as suas respectivas associações com os casos de uso. De forma mais clara, cada ator só pode executar os requisitos funcionais com os quais estão associados. Na seção 4.2.1 são descritos em detalhes cada requisito funcional.

Figura 4.1: Casos de uso do ambiente computacional



powered by Astah

Fonte: Autor, 2013.

²“Um ator é um classificador que representa os objetos externos ao sistema modelado (papéis de usuários, outros sistemas, etc) que interagem com um caso de uso.” (Paula Filho, 2010)

Descrição dos casos de uso

Gestão de pesquisadores e colaboradores

Descrição: Controle de pesquisadores e colaboradores que possuem acesso ao ambiente. Provê funcionalidades de inclusão, consulta, atualização e exclusão.

Pré-condições: Configuração de todos os parâmetros do ambiente.

Pós-condições: Pesquisadores/colaboradores associados aos projetos de pesquisa.

Gestão de projetos de pesquisa

Descrição: Controle dos projetos de pesquisa gerenciados pelo ambiente. Provê funcionalidades de inclusão, consulta, atualização e exclusão.

Pré-condições: Configuração de todos os parâmetros do ambiente.

Pós-condições: Permissão para gerar uma versão do projeto de pesquisa.

Geração de nova versão do projeto de pesquisa

Descrição: Criação de uma nova versão do projeto de pesquisa.

Pré-condições: Projeto de pesquisa cadastrado no ambiente.

Pós-condições: Permissão para a geração da página web do projeto de pesquisa.

Gestão da configuração do ambiente

Descrição: Controle das configurações obrigatórias do ambiente, com consulta e atualização de parâmetros.

Pré-condições: Ambiente instalado.

Pós-condições: Permissão para cadastro de pesquisadores, colaboradores e projetos de pesquisa.

Gestão dos artefatos do projeto de pesquisa

Descrição: Controle dos artefatos de um projeto de pesquisa. Provê inclusão, consulta, atualização e exclusão.

Pré-condições: Projeto de pesquisa cadastrado no ambiente.

Pós-condições: Artefato disponível.

Gestão dos dados pessoais

Descrição: Controle dos dados pessoais do pesquisador ou colaborador, com consulta e atualização de parâmetros.

Pré-condições: pesquisadores/colaboradores cadastrados no ambiente.

Pós-condições: dados pessoais dos pesquisadores/colaboradores alterados.

Geração da página web do projeto de pesquisa

Descrição: Criação ou atualização de uma página web para o projeto de pesquisa.

Pré-condições: Versão de um projeto de pesquisa cadastrada no ambiente.

Pós-condições: Permissão de acesso público para todos os artefatos de um projeto de pesquisa.

Os casos de uso formam o ponto de partida para, especificamente neste trabalho, determinar classes, atributos e operações, que serão mostrados na seção 4.3.

4.2.2 Modelagem da base de dados

Trata-se de uma das etapas mais importantes para o desenvolvimento de um sistema de software. De forma sucinta, essa atividade consiste em representar o ambiente observado, documentar e normalizar, fornecer processos de validação e observar processos de relacionamentos entre os objetos do sistema a ser desenvolvido.

Como ferramentas de modelagem de dados, são escolhidos o modelo conceitual e o modelo lógico. O modelo conceitual é baseado no conjunto de objetos, relevantes no domínio do sistema modelado, chamado entidades, e pelo conjunto de relacionamentos entre esses objetos. O modelo lógico leva em consideração algumas limitações do banco de dados relacional e compreende recursos de padrão, nomenclatura, chaves primárias³ e estrangeiras⁴, normalização, integridade referencial, dentre outros.

Na figura 4.2 é apresentado o modelo conceitual do ambiente computacional proposto. De forma visual, temos que:

Retângulos: Representam as entidades relevantes no ambiente;

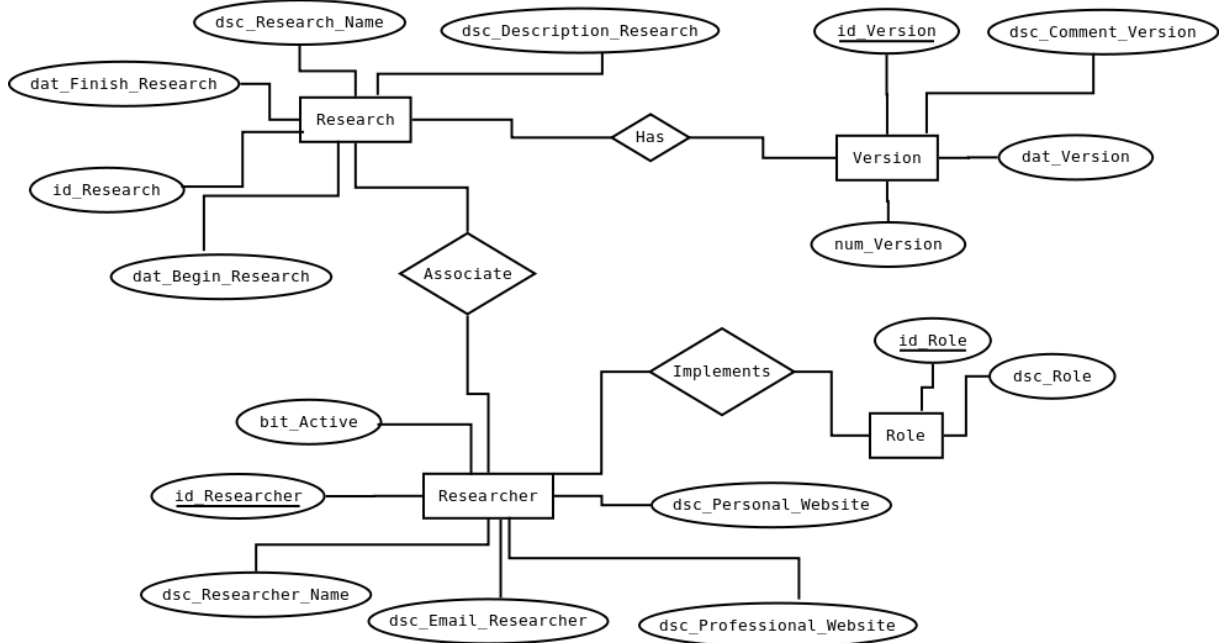
Elipses: Representam os atributos mais importantes em cada entidade representada;

Losangos: Representam as associações entre as entidades do ambiente.

³Chave primária é o conjunto de um ou mais atributos escolhidos pelo projetista do banco de dados que, tomados coletivamente, permitem identificar uma instância de uma tabela. (Silberschatz et al., 1999)

⁴Chave estrangeira é o conjunto de um ou mais atributos existentes em uma tabela A, que é chave primária em uma tabela B.

Figura 4.2: Modelo Entidade-Relacionamento do ambiente computacional



Fonte: Autor, 2013.

Na figura 4.3 é apresentado o modelo lógico do ambiente computacional. Tal modelo foi desenvolvido a partir das entidades, atributos e associações definidas no modelo conceitual. De forma gráfica temos que:

Retângulos: Representam as tabelas e as colunas para um banco de dados relacional. Em especial, a(s) coluna(s) sublinhada(s) representam a chave primária da tabela;

Setas: Representam as associações entre as tabelas. Quando uma tabela possui uma seta na sua direção, significa que ela contém uma coluna, denominada chave estrangeira, referenciada como chave primária em outra tabela, por exemplo, a tabela *Research* contém a coluna *id_role* que é chave primária da tabela *Role*.

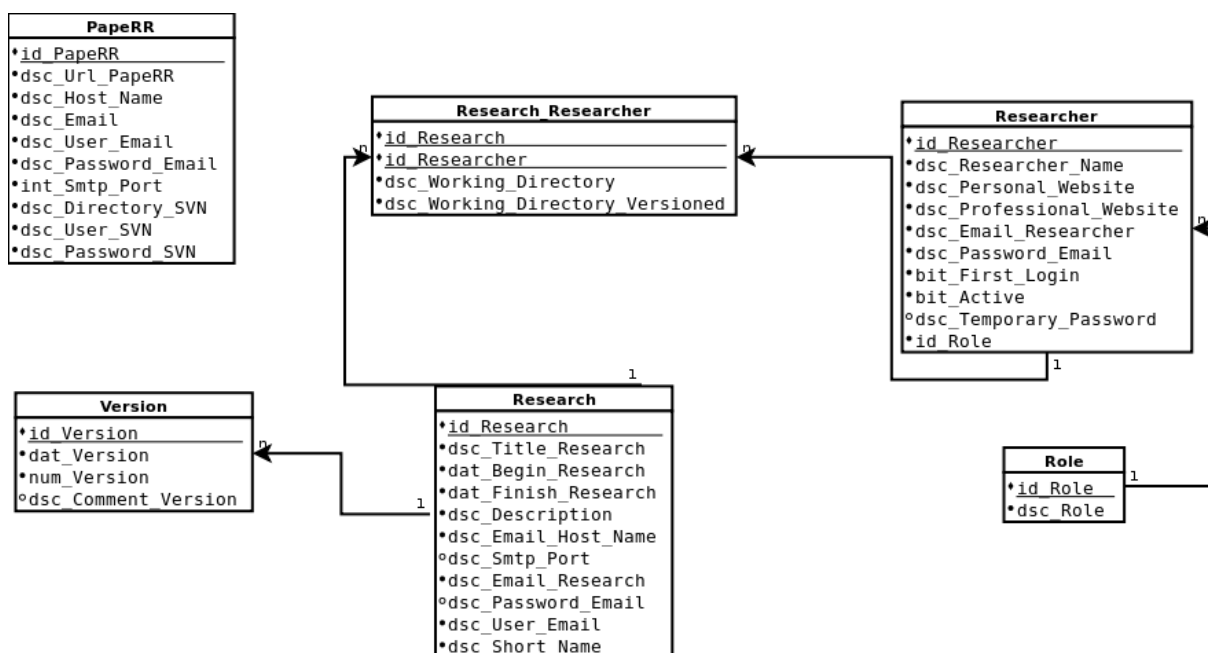


Figura 4.3: Modelo lógico do ambiente computacional

Dicionário de dados

O dicionário de dados descreve e define o significado de todos os dados que compõem o universo de informações de um sistema. Sua principal utilidade, no contexto deste trabalho, é a de fornecer e padronizar definições semânticas a serem adotadas em todo o sistema. Um dicionário de dados atualizado possibilita facilidades para a manutenção e extensão do sistema, bem como proporciona uma documentação da base de dados utilizada. Logo abaixo, de forma resumida, é apresentado o dicionário de dados que contém a descrição de todas as tabelas do sistema.

PapeRR: Tabela onde são armazenados todos os parâmetros do ambiente computacional.

Research: Tabela onde são armazenados todos os projetos de pesquisa.

Researcher: Tabela onde são armazenados todos os pesquisadores/colaboradores.

Research_Researcher: Tabela onde são armazenadas as associações entre projetos de pesquisa e seus pesquisadores/colaboradores.

Role: Tabela onde são armazenados os papéis/funções que os pesquisadores/colaboradores podem desempenhar no ambiente.

Version: Tabela onde são armazenadas as versões dos projetos de pesquisa.

4.3 Projeto do ambiente computacional

Nesta seção são apresentados a arquitetura, os diagramas de classes e os diagramas de atividades que compõem o projeto do ambiente computacional.

4.3.1 Arquitetura do ambiente computacional

De forma sucinta, a arquitetura de um sistema define a forma que o código fonte é organizado. No caso do ambiente computacional proposto, a decisão recaiu sobre a implementação em camadas. Esta prática, de forma geral, torna o código fonte mais legível, além de aumentar a flexibilidade e escalabilidade do sistema. Em termos de arquitetura, o sistema possui as seguintes camadas:

view: Camada onde se encontram os componentes responsáveis pela interface visual.

service: Camada onde se encontram os componentes pela mediação entre a camada *view* e as outras camadas.

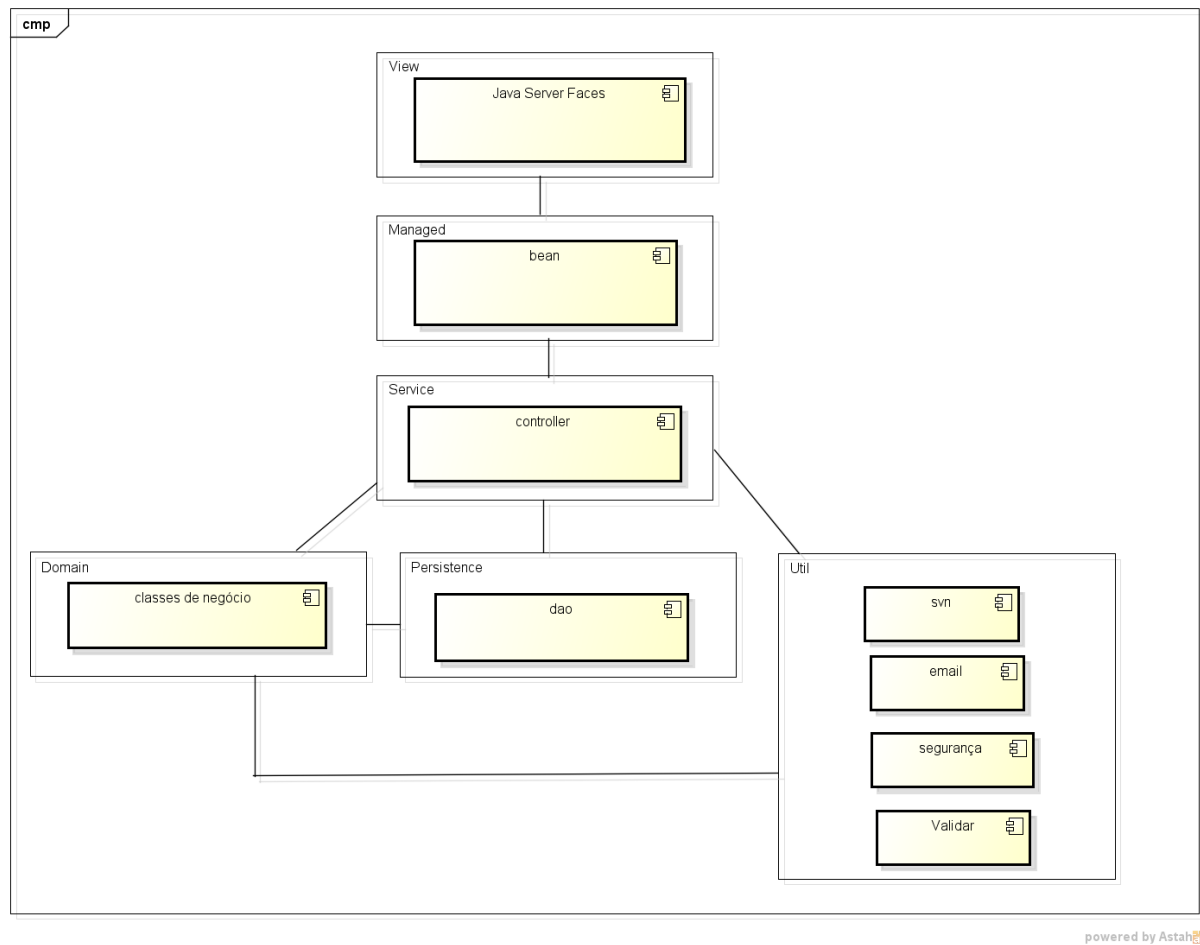
domain: Camada onde se encontram os componentes do domínio do sistema.

persistence: Camada onde se encontram os componentes responsáveis pela mediação com o banco de dados.

util: Camada onde se encontram os componentes responsáveis pelas ações de envio de email, criptografia, segurança e integração com outros softwares.

Na Figura 4.4 é apresentado o diagrama com as camadas do ambiente. Os retângulos maiores representam as camadas existentes no ambiente. Em cada camada, é possível observar os seus componentes ou conjunto de classes, que são responsáveis por determinadas funcionalidades. Além disso, cada linha no diagrama representa uma associação existente entre as classes dessas camadas. Maiores detalhes das associações são encontradas nas figuras 4.5, 4.6 e 4.7.

Figura 4.4: Camadas do ambiente computacional



Fonte: Autor, 2013.

4.3.2 Diagrama de classes

Um diagrama de classes é uma representação de estrutura e relações das classes que devem compor o sistema de software. No caso do ambiente computacional, todos os diagramas de classes mostrados apresentam uma visão mais detalhada das camadas mais importantes (*domain*, *persistence* e *service*), bem como constituem-se em base para a construção dos diagramas de atividades que estão dispostos na seção 4.3.3.

Na Figura 4.5 é apresentado o diagrama de classes do pacote *domain*. As classes listadas abaixo são constituintes do pacote *domain*.

Research: Classe responsável pelos atributos e operações do objeto *research*.

Researcher: Classe responsável pelos atributos e operações do objeto *researcher*.

Research_Researcher: Classe responsável pelos atributos e operações do objeto *research_researcher*.

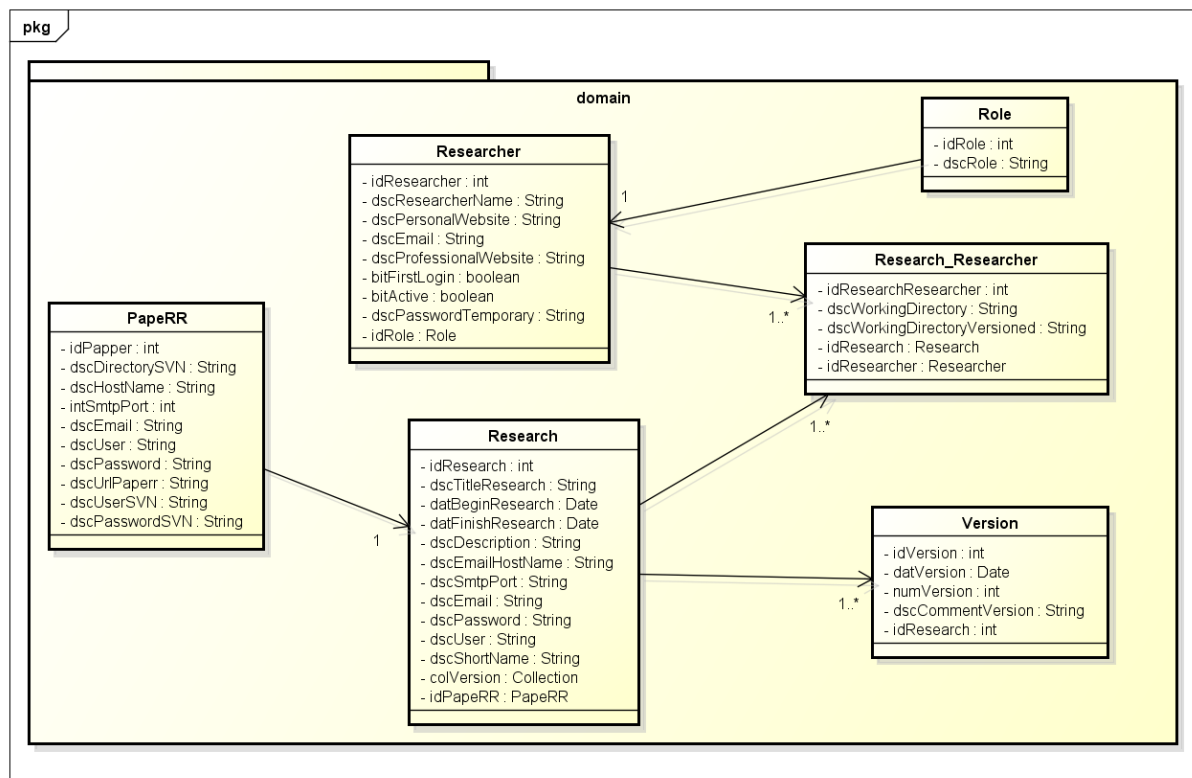
Role: Classe responsável pelos atributos e operações do objeto *role*.

Version: Classe responsável pelos atributos e operações do objeto *version*.

PapeRR: Classe responsável pelos atributos e operações do objeto *paperr*.

Uma classe pode ser estruturada em até três compartimentos: o primeiro lista o nome da classe, o segundo lista os atributos e o terceiro lista as operações. Neste diagrama, optou-se por exibir a classe com dois compartimentos (nome da classe e atributos), para cada atributo temos o seu nome e tipo (inteiro, *booleano*, *string*, dentre outros). As setas representam associações entre as classes, na ponta de cada seta é apresentado o limite inferior e superior da quantidade de objetos na qual um outro objeto pode estar associado.

Figura 4.5: Diagrama de classes do pacote *domain*



powered by Astah

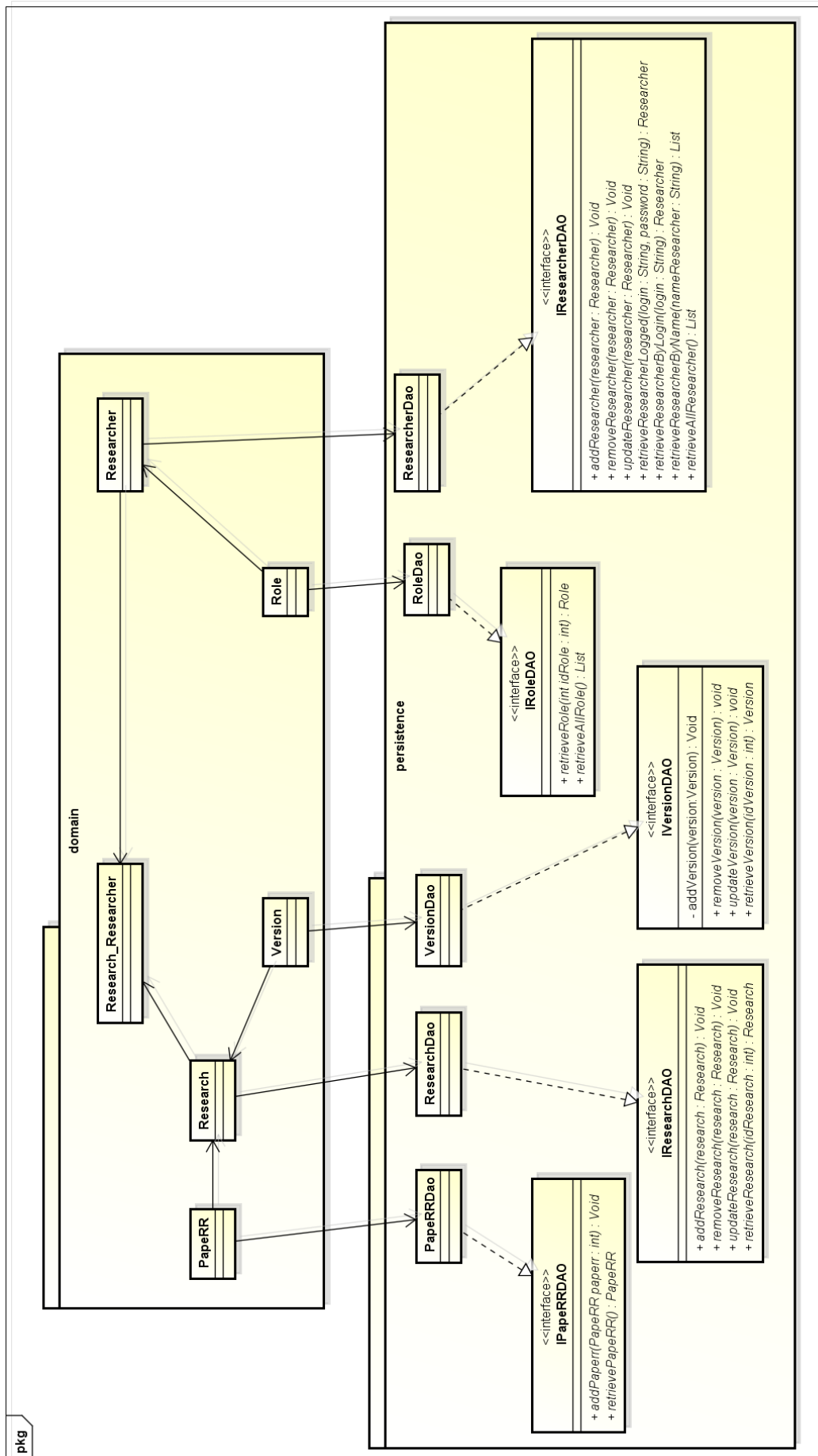
Fonte: Autor, 2013.

Na Figura 4.6 é apresentado o diagrama de classes do pacote *persistence*. Todas as classes desse pacote estão associadas à sua respectiva classe no pacote *domain* e implementa uma interface⁵ específica. Por exemplo, temos que a classe *VersionDao* está associada à classe

⁵Uma interface contém valores constantes ou assinaturas de métodos que devem ser implementados dentro de uma classe.

Version implementa a interface *IVersionDao*. Optou-se, nesse caso, pela exibição da classe com os compartimentos de nome da classe e operações.

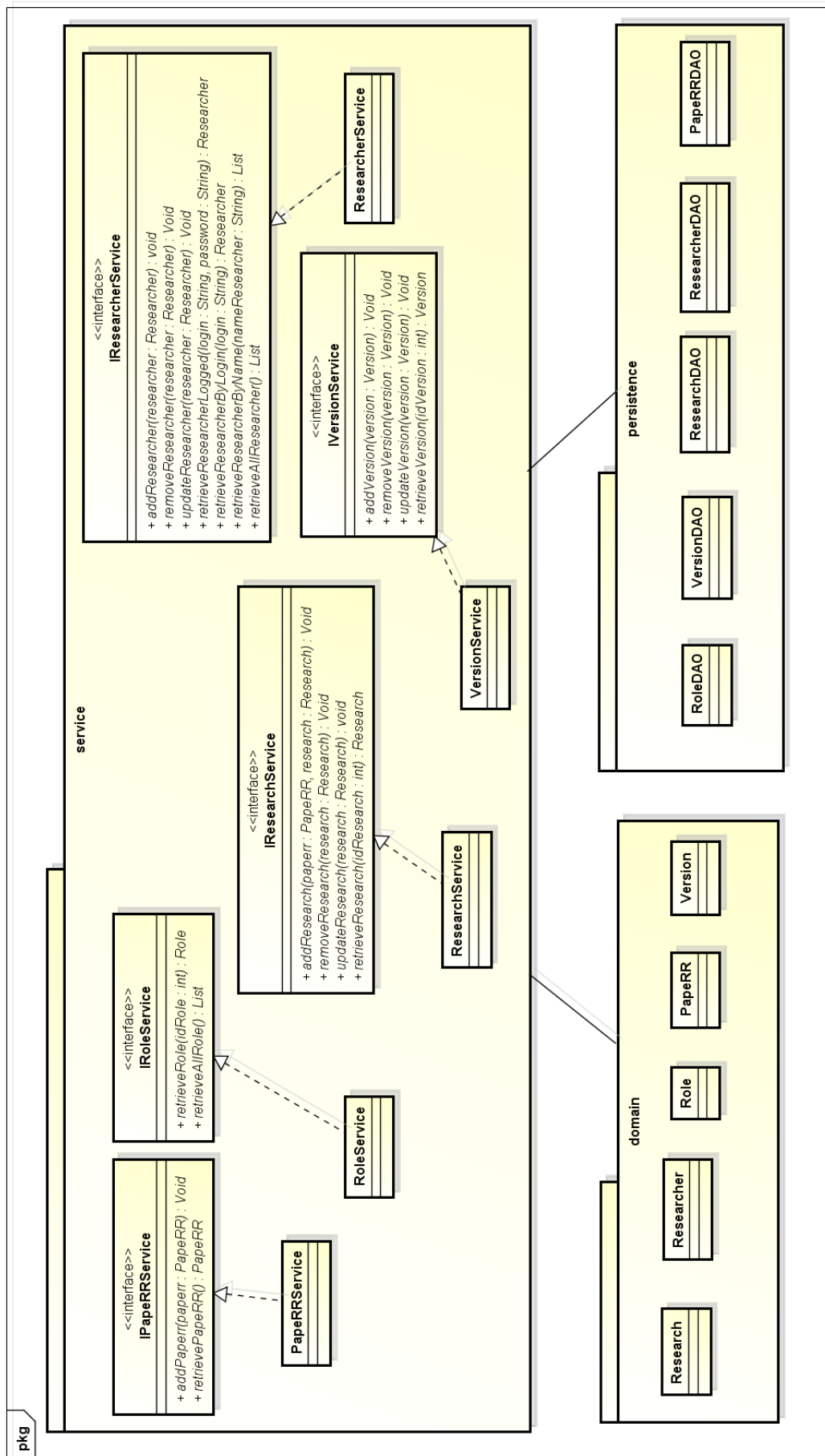
Figura 4.6: Diagrama de classes do pacote *persistence*



powered by Astah

Na Figura 4.7 é apresentado o diagrama de classes do pacote *service*. Todas as classes desse pacote estão associadas à sua respectiva classe no pacote *domain* e no pacote *persistence*, e implementa uma interface específica. Optou-se, nesse caso, pela exibição da classe com os compartimentos de nome da classe e operações.

Figura 4.7: Diagrama de classes do pacote *service*



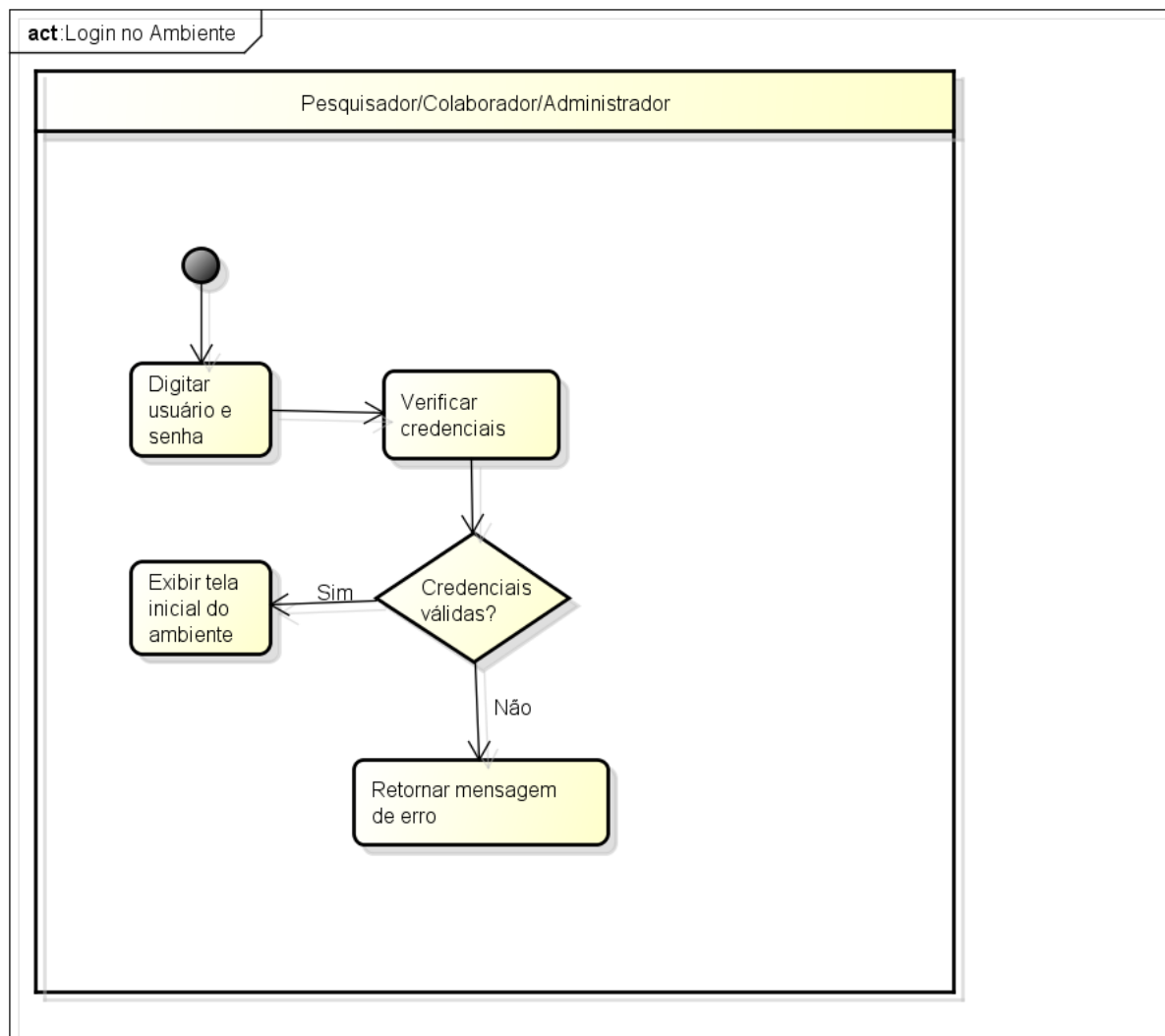
powered by Astah

4.3.3 Diagrama de atividades

Os diagramas de atividades têm o objetivo de representar uma especificação de comportamento executável, em particular no ambiente computacional. O principal intuito é mostrar as ações, sem especificar detalhes, das principais atividades que acontecem durante a execução de um caso de uso do ambiente computacional.

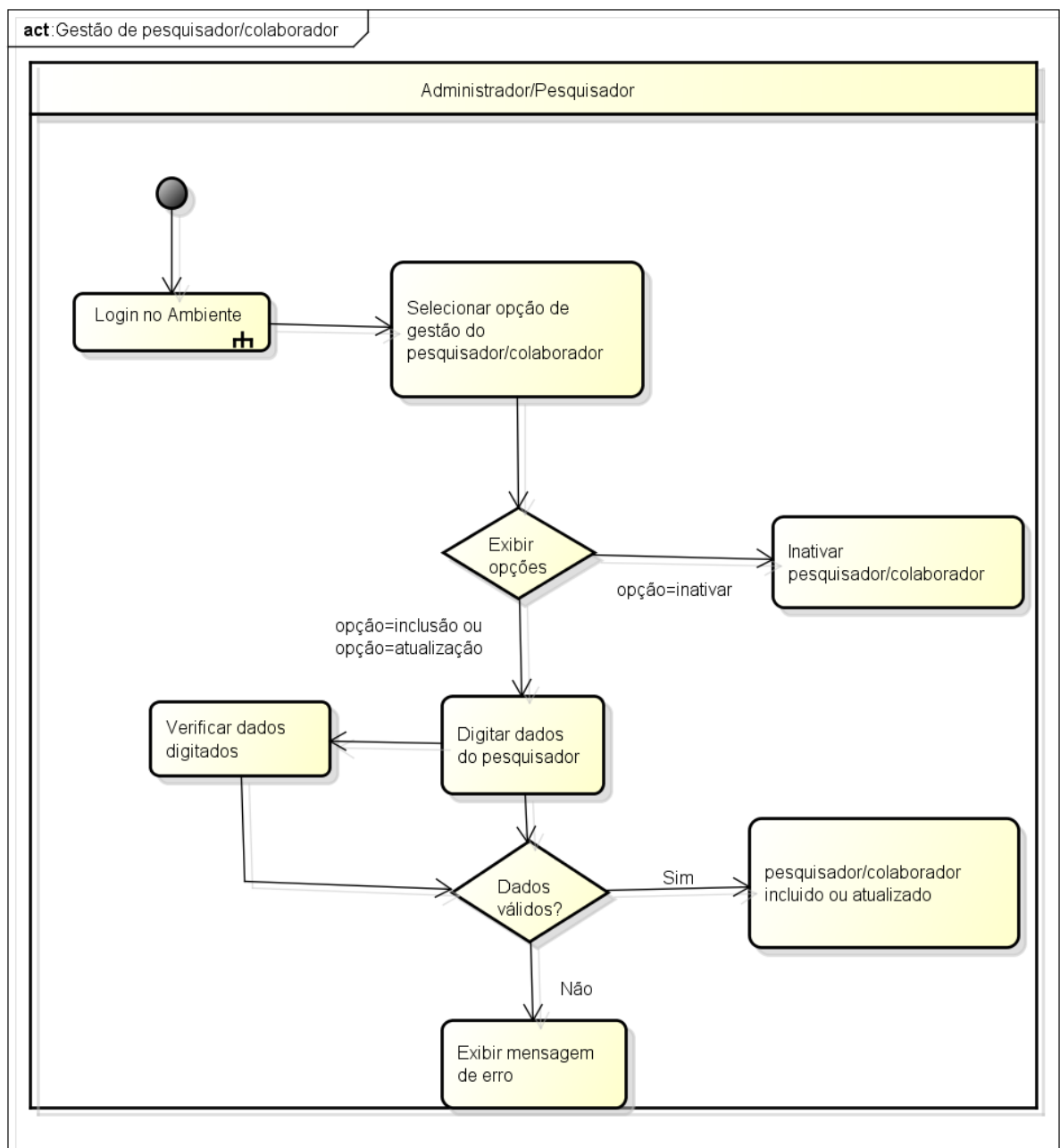
Na Figura 4.8 são apresentadas as ações que ocorrem durante o caso de uso Efetuar Login no Ambiente. Neste caso de uso o ator (Pesquisador, Colaborador ou Administrador) digita as credenciais de acesso ao sistema. Caso as credenciais sejam válidas, é exibida a tela inicial do sistema, caso contrário, o sistema exibe uma mensagem de erro.

Figura 4.8: Diagrama de atividades relacionado ao login no ambiente



Na Figura 4.9 são exibidas as ações que ocorrem durante o caso de uso Gestão de pesquisador/colaborador. Neste caso de uso o ator (Administrador ou Pesquisador) seleciona a opção de gestão de pesquisador/colaborador na tela inicial do ambiente, com a intenção de inativar, atualizar ou incluir um pesquisador/colaborador.

Figura 4.9: Diagrama de atividades relacionado à gestão de pesquisadores e colaboradores.



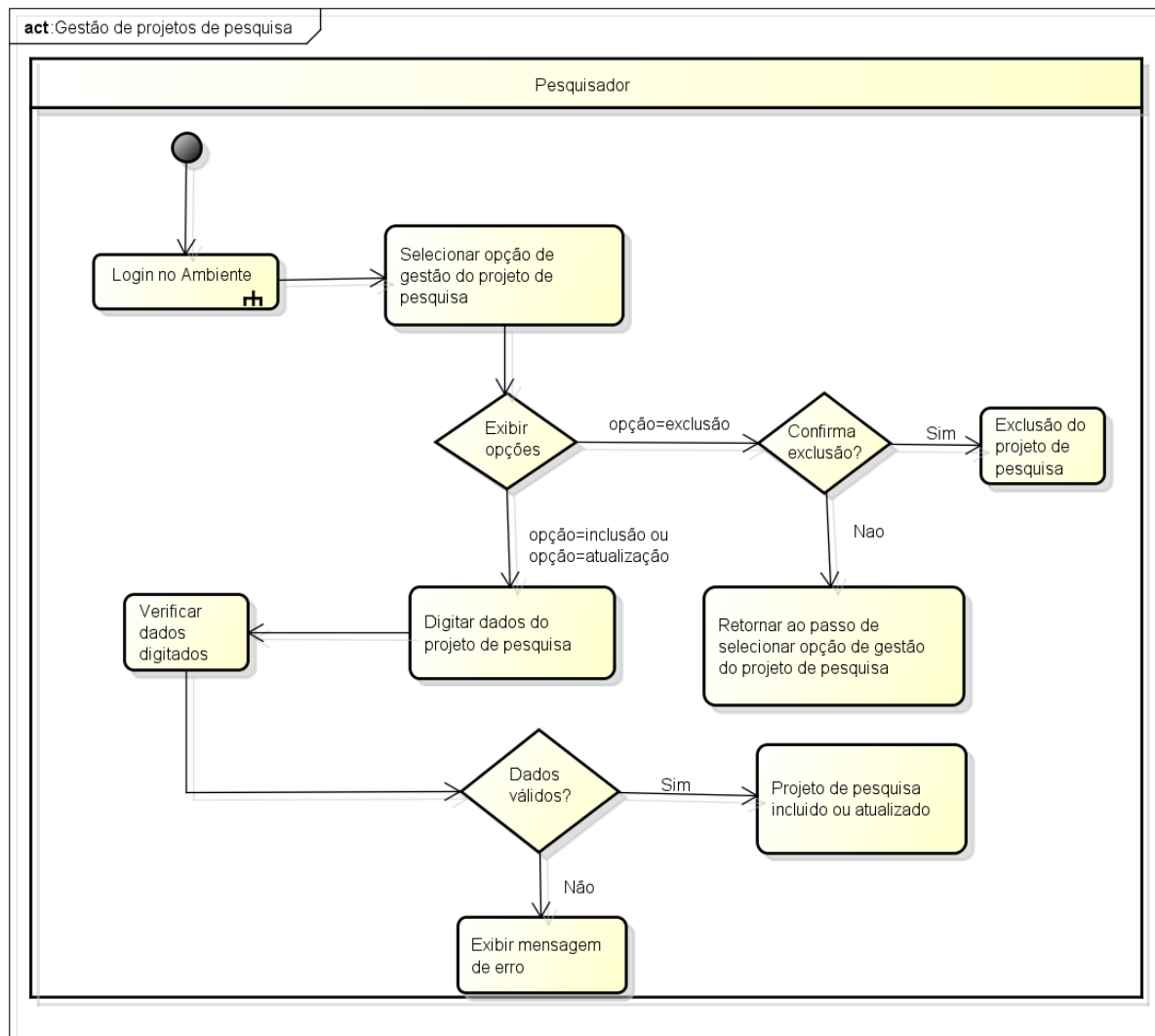
powered by Astah

Fonte: Autor, 2013.

Na Figura 4.10 são exibidas as ações que ocorrem durante o caso de uso Gestão de projeto de pesquisa. Neste caso de uso o ator (Pesquisador) seleciona um projeto ao qual está

associado, com a intenção de excluir, atualizar ou incluir um projeto de pesquisa.

Figura 4.10: Diagrama de atividades relacionado à gestão de projetos de pesquisa

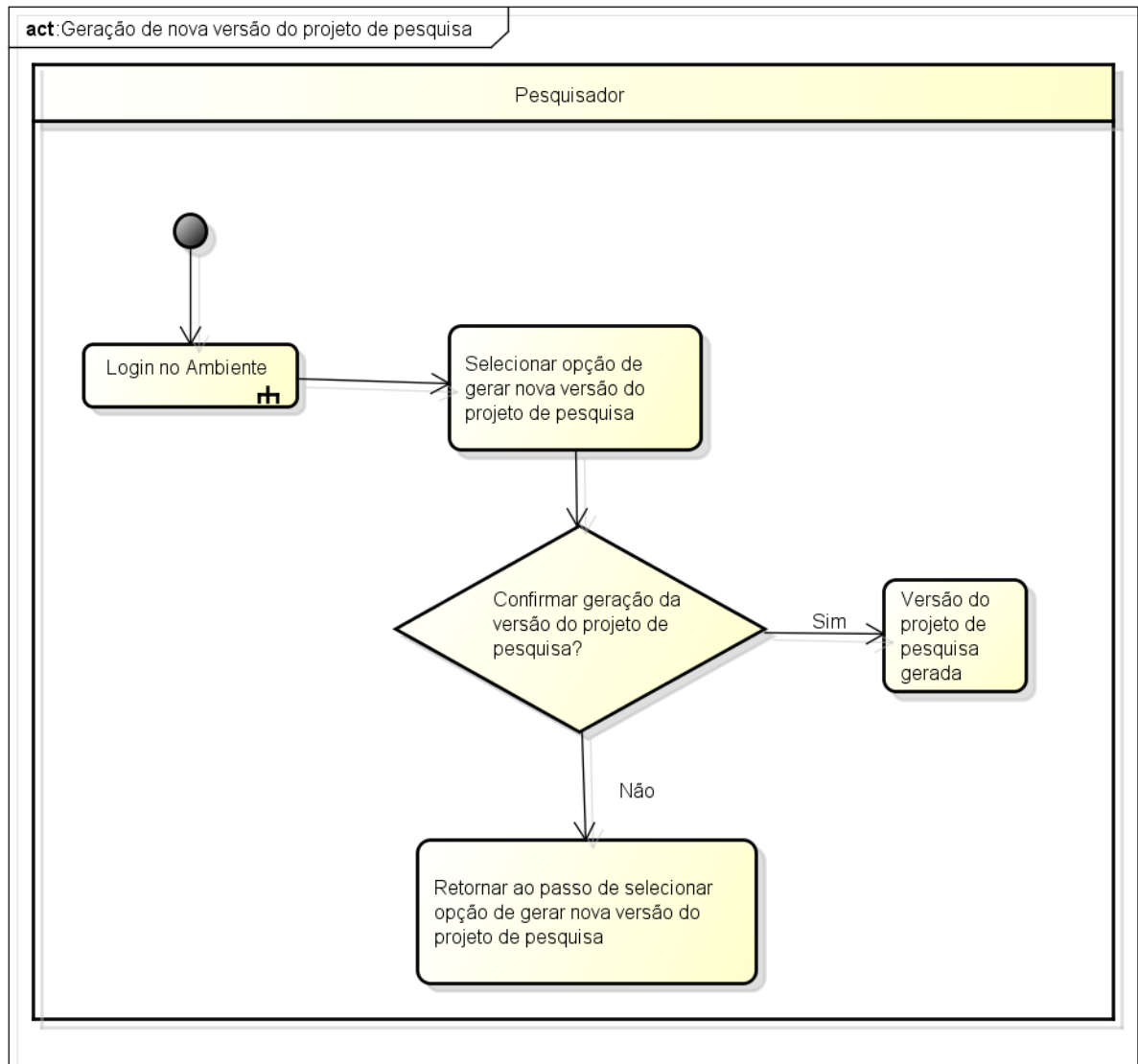


powered by Astah

Fonte: Autor, 2013.

Na Figura 4.11 são exibidas as ações que ocorrem durante o caso de uso Gestão de versão de um projeto de pesquisa. Neste caso, o ator (Pesquisador) seleciona um projeto ao qual está associado, com a intenção de gerar uma nova versão desse projeto.

Figura 4.11: Diagrama de atividades relacionado à gestão de versão de um projeto de pesquisa

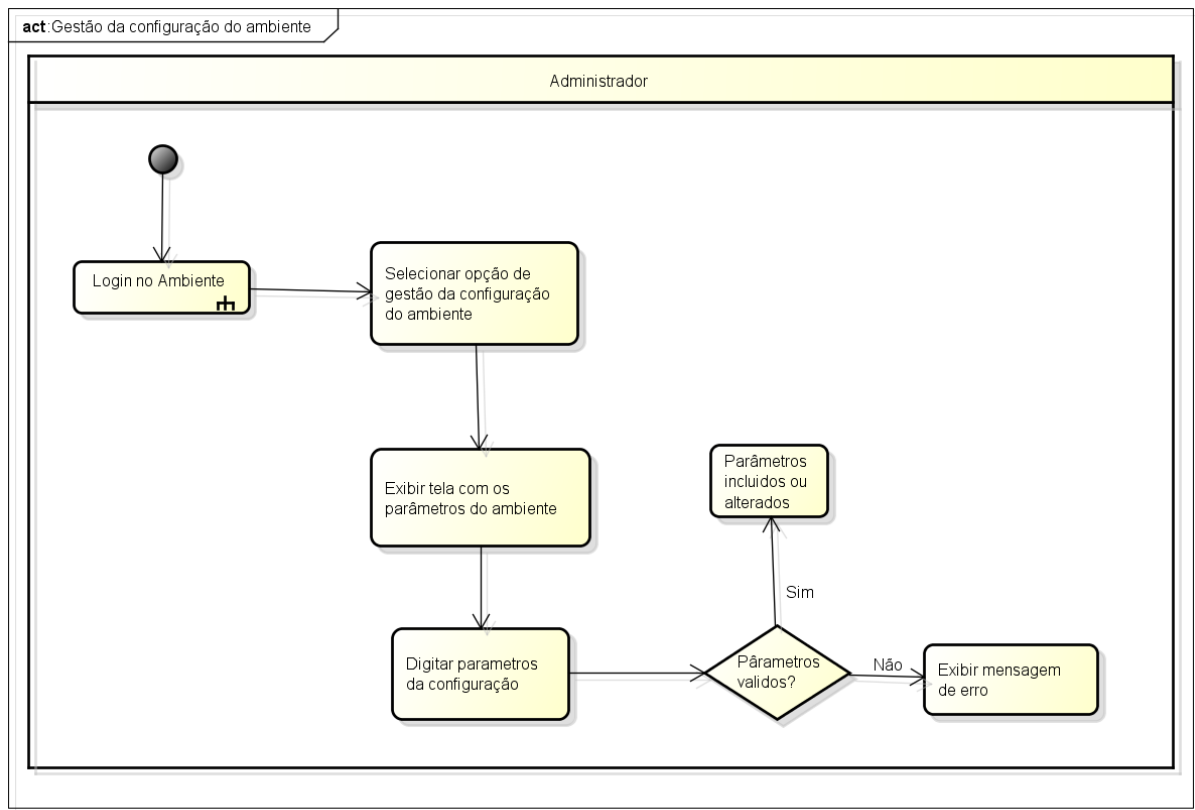


powered by Astah

Fonte: Autor, 2013.

Na Figura 4.12 são exibidas as ações que ocorrem durante o caso de uso Gestão de configuração do ambiente. Neste caso, o ator (Administrador) pode incluir ou alterar as configurações do ambiente computacional.

Figura 4.12: Diagrama de atividades relacionado à gestão da configuração do ambiente

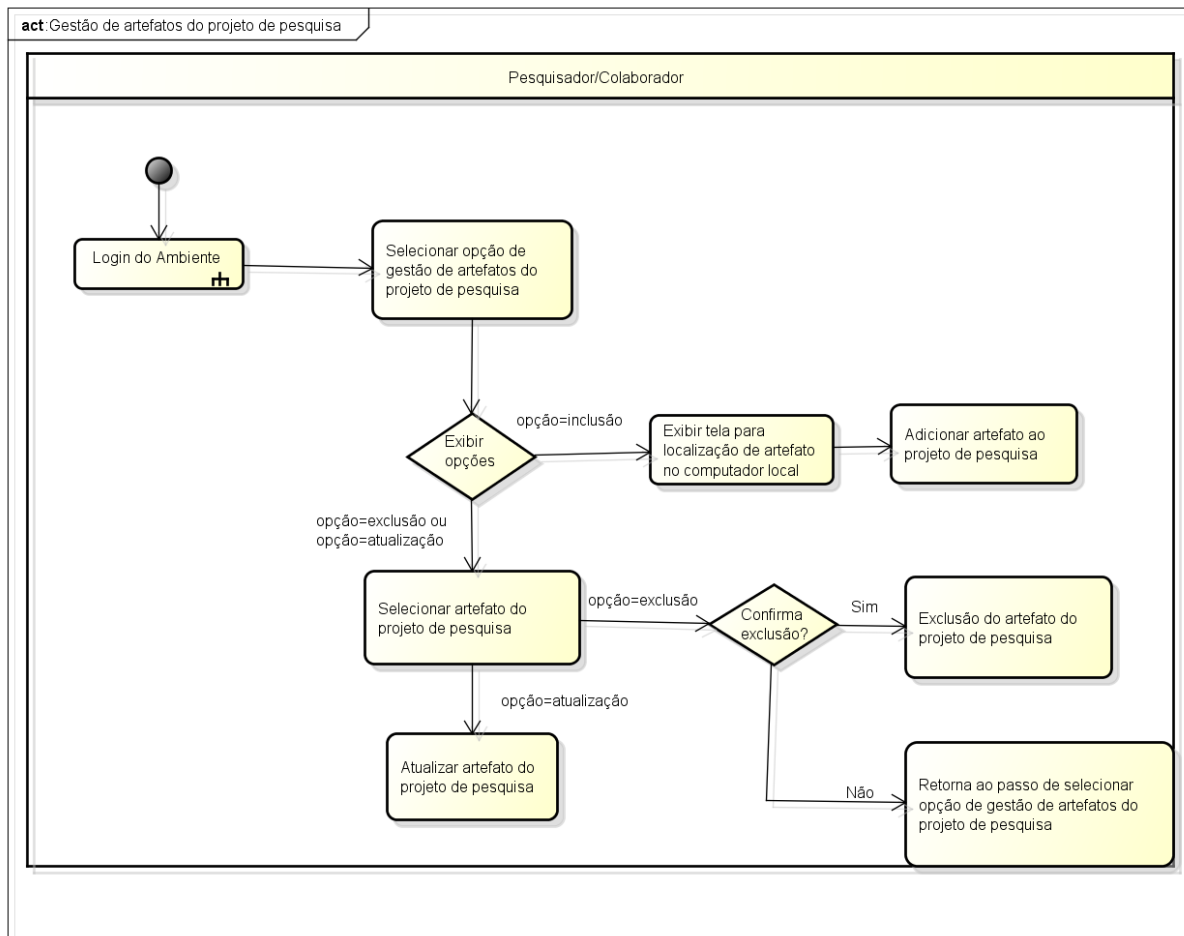


powered by Astah

Fonte: Autor, 2013.

Na Figura 4.13 são exibidas as ações que ocorrem durante o caso de uso Gestão de artefatos do projeto de pesquisa. Neste caso, o ator (Pesquisador ou Colaborador) seleciona um projeto ao qual está associado, com a intenção de incluir, atualizar ou excluir um artefato deste projeto.

Figura 4.13: Diagrama de atividades relacionado à gestão de artefatos do projeto de pesquisa

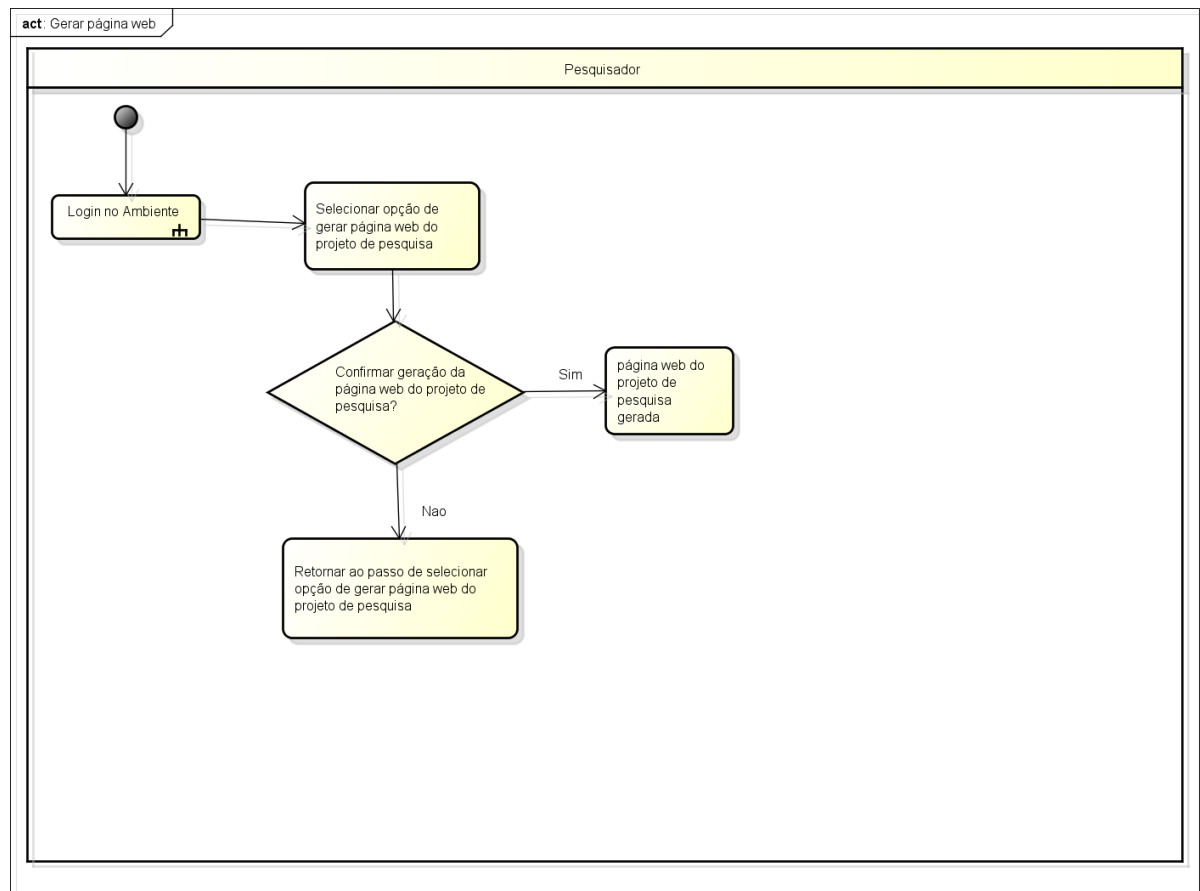


powered by Astah

Fonte: Autor, 2013.

Na Figura 4.14 são exibidas as ações que ocorrem durante o caso de uso geração da página web de um projeto de pesquisa. Neste caso, o ator (Pesquisador) seleciona um projeto ao qual está associado, com a intenção de gerar uma página web para o projeto.

Figura 4.14: Diagrama de atividades relacionado à geração da página web de um projeto de pesquisa



powered by Astah

Fonte: Autor, 2013.

O capítulo apresentou os modelos e diagramas do ambiente computacional. No próximo capítulo será apresentado o ambiente computacional.

5

Ambiente computacional proposto

Neste capítulo são apresentados os passos para a instalação do ambiente computacional e as suas principais funcionalidades.

5.1 Preparação do ambiente computacional

Nas seções a seguir são detalhados os passos necessários para a instalação e configuração do ambiente computacional.

5.1.1 Instalação e configuração do ambiente computacional

O ambiente computacional foi implantado em um notebook com características a seguir:

- Sistema operacional *Ubuntu* na versão 10.04;
- Processador *Intel Core i5* com 4GB de memória RAM;
- Banco de dados *MySQL* na versão 5.2.28;
- Navegadores web *Mozilla Firefox* na versão 20.0 e *Google Chrome* na versão 27.0;
- (?)

Os softwares listados a seguir devem ser instalados:

- Banco de dados *MySQL*, que pode ser obtido no endereço <http://dev.mysql.com/downloads/>;
- Servidor web *Apache Tomcat*, que pode ser obtido no endereço <http://tomcat.apache.org/>;

Após a instalação do banco de dados *MySQL*, é necessário criar a base de dados do ambiente computacional. Nesta etapa, há o cadastro de todas as tabelas, bem como a inserção dos usuários administrador da base de dados e administrador do ambiente computacional. Para concluir a instalação do ambiente, é preciso colocar o arquivo *Reproducible_web.war* no diretório *webapps* da instalação do *Apache Tomcat*. Maiores detalhes sobre a instalação do ambiente computacional podem ser obtidos na <https://sites.google.com/site/projetoreproducibleresearch/>.

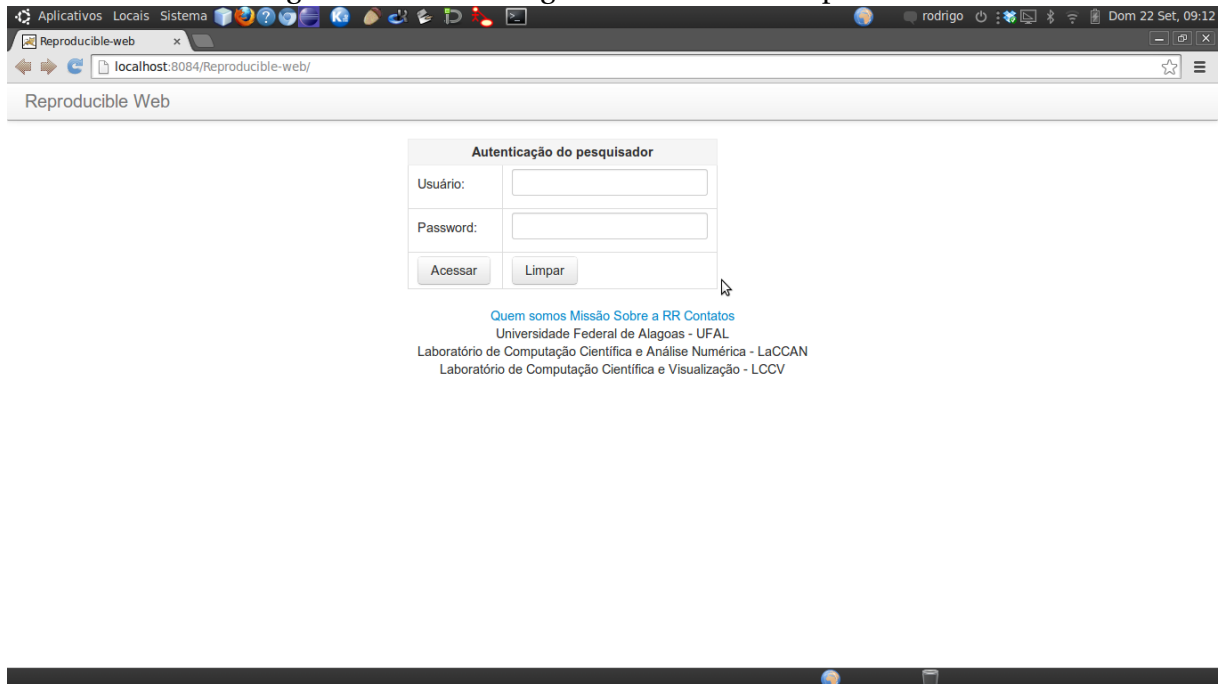
5.2 Principais funcionalidades do ambiente computacional

Seguem as principais funcionalidades do ambiente computacional:

5.2.1 Efetuar login

Essa funcionalidade refere-se à autenticação do usuário (pesquisador, colaborador ou administrador) no ambiente computacional. A seguir, a figura 5.1 mostra a tela de login do ambiente computacional.

Figura 5.1: Tela de login do ambiente computacional

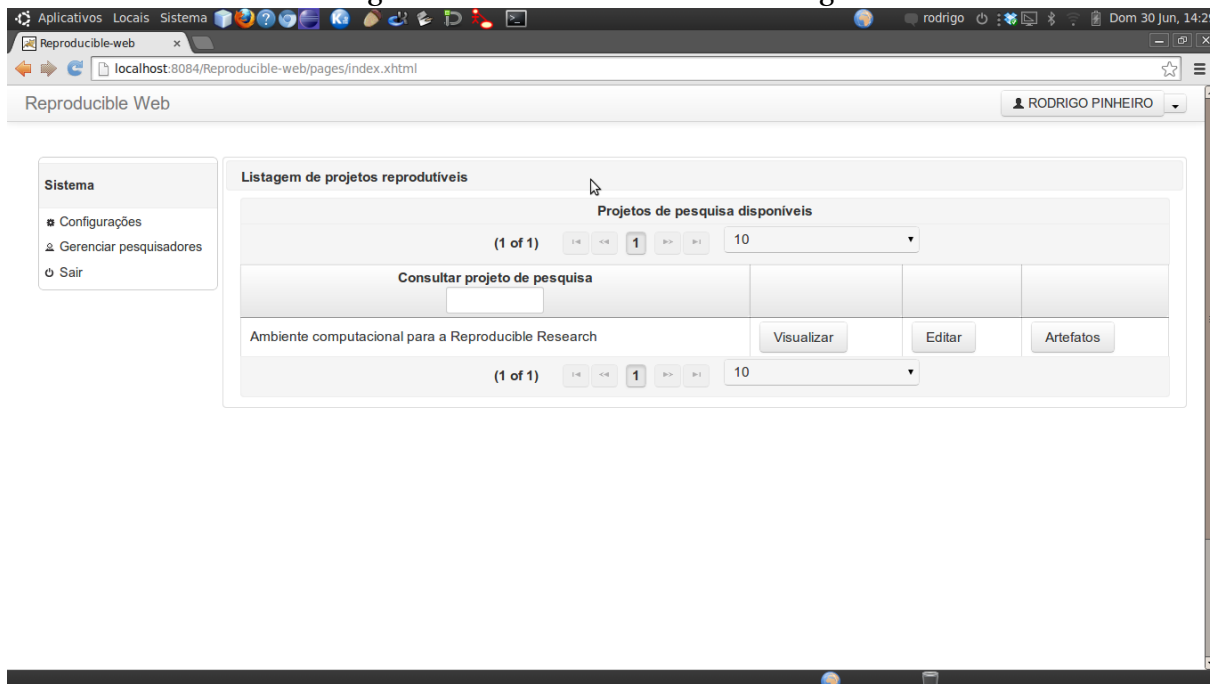


Fonte: Autor, 2013.

Após a digitação das credenciais (usuário e *password*) o ambiente verifica se as mesmas são válidas, caso positivo é exibida a tela inicial do usuário, caso contrário o ambiente retorna uma mensagem de erro.

Caso as credenciais sejam válidas, é exibida para o usuário a tela inicial, que lista os projetos de pesquisa aos quais o usuário está associado, como mostra a figura 5.2. Vale destacar que não é possível associar um projeto de pesquisa a um usuário com papel de administrador.

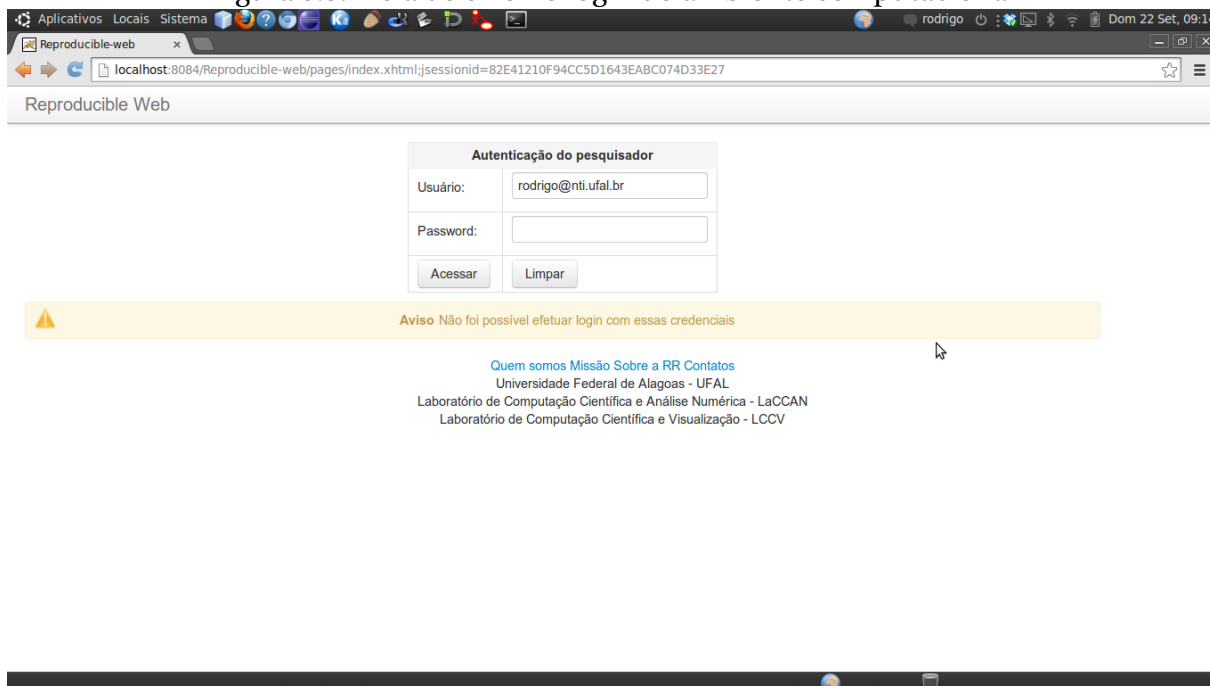
Figura 5.2: Tela inicial do usuário logado



Fonte: Autor, 2013.

Caso as credenciais não sejam válidas, é exibida para o usuário a tela conforme a figura abaixo 5.3:

Figura 5.3: Tela de erro no login do ambiente computacional



Fonte: Autor, 2013.

5.2.2 Gestão de pesquisador/colaborador

Essa funcionalidade refere-se à gestão (incluir, alterar, recuperar e inativar) de pesquisadores e colaboradores. Para ter acesso a essas ações, é necessário possuir os papéis de administrador ou pesquisador. Na figura 5.4 é exibida a tela com a listagem de todos os pesquisadores cadastrados no ambiente computacional.

Figura 5.4: Tela com a listagem dos pesquisadores cadastrados no ambiente computacional

Reproducible Web

RODRIGO PINHEIRO

Sistema

- Configurações
- Gerenciar pesquisadores
- Sair

Listagem de pesquisadores

(1 of 1) 1 10

Nome do pesquisador	Email	Site profissional	
RODRIGO PINHEIRO	rodrigo@nti.ufal.br	http://lattes.cnpq.br/0365438845525046	Detalhes
THAYSE DOS SANTOS FONSECA	thaysefonseca16@gmail.com	http://lattes.cnpq.br/7227645365126862	Detalhes
ALEJANDRO C. FRERY	acfrery@gmail.com	http://lattes.cnpq.br/6935433850568144	Detalhes

(1 of 1) 1 10

+ Novo Pesquisador Home

Quem somos Missão Sobre a RR Contatos
Universidade Federal de Alagoas - UFAL
Laboratório de Computação Científica e Análise Numérica - LaCCAN
Laboratório de Computação Científica e Visualização - LCCV

Fonte: Autor, 2013.

A partir da tela de listagem de pesquisadores, é possível acessar os dados cadastrais através da opção detalhes, conforme mostra a figura 5.5:

Figura 5.5: Tela que exhibe os detalhes do cadastro de um pesquisador no ambiente computacional



Fonte: Autor, 2013.

A condição de inativo aplicada a um determinado usuário, impossibilita-o de se autenticar no ambiente e de ser associado a um projeto de pesquisa.

A figura 5.6 a seguir mostra a tela de cadastro de pesquisador/colaborador. Ao final do cadastro é obrigatória a definição do papel que o pesquisador/colaborador terá no ambiente computacional.

Figura 5.6: Tela de cadastro do pesquisador

The image shows a web browser window with the following elements:

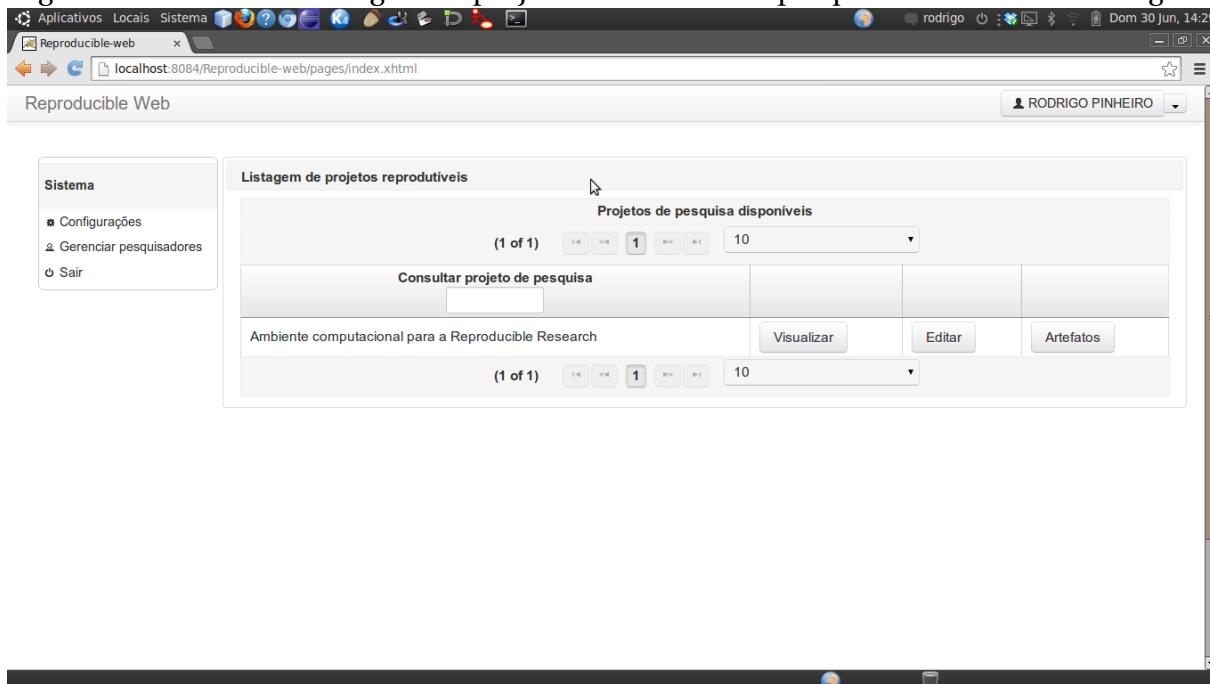
- Browser tabs: 'Reproducibile-web' (two tabs).
- Address bar: 'localhost:8084/Reproducibile-web/pages/listResearcher.xhtml'.
- Page title: 'Reproducibile Web'.
- User profile: 'RODRIGO PINHEIRO'.
- Left sidebar menu:
 - Sistema
 - Configurações
 - Gerenciar pesquisadores
 - Sair
- Main content area: 'Cadastro de pesquisador' form.
 - Nome:
 - Email:
 - Site web profissional:
 - Site web pessoal:
 - Papel do Pesquisador:
 - Buttons: Salvar, Limpar, Home

Fonte: Autor, 2013.

5.2.3 Gestão de projeto de pesquisa

Essa funcionalidade refere-se à gestão (incluir, alterar e recuperar) do projeto de pesquisa. Para ter acesso a essa ação é necessário possuir o papel de pesquisador ou colaborador.

Figura 5.7: Tela com a listagem de projetos associados ao pesquisador/colaborador logado



Fonte: Autor, 2013.

Na figura 5.7 são mostrados todos os projetos aos quais o pesquisador/colaborador está associado. Também é possível ter acesso às opções de visualizar, editar e artefatos. A opção de visualizar exibe, sem a possibilidade de edição, os dados cadastrais do projeto de pesquisa. Logo abaixo, segue a figura 5.8 que exibe os dados básicos do projeto de pesquisa:

Figura 5.8: Tela de visualização dos dados básicos do projeto de pesquisa

Reproducibile Web

RODRIGO PINHEIRO

Sistema

- Configurações
- Gerenciar pesquisadores
- Sair

Dados Básicos

Titulo da Pesquisa: *	Ambiente computacional para suporte a Reproducibile Researcher
Short name: *	reproducibile
Data de Inicio do projeto: *	11/00/2013
Descrição do projeto: *	Reproducibile Researcher

Fonte: Autor, 2013.

Na figura 5.9 são exibidos os dados de configuração do projeto de pesquisa. Esses dados são utilizados para as ações relacionadas ao envio de email aos pesquisadores/colaboradores vinculados ao projeto.

Figura 5.9: Tela de visualização das configurações do projeto de pesquisa

Reproducibile Web

RODRIGO PINHEIRO

Sistema

- Configurações
- Gerenciar pesquisadores
- Sair

Configurações

Endereço do servidor de SMTP:	smtp.gmail.com
Porta do servidor SMTP	625
Endereço de email do projeto	rodrigolpinheiro@gmail.com
Usuário de email do projeto	rodrigolpinheiro
Senha do email do projeto	

Fonte: Autor, 2013.

Na figura 5.10 é exibida a tela que visualiza a associação entre pesquisadores/colaboradores e um projeto de pesquisa.

Figura 5.10: Tela de visualização da associação entre pesquisadores e projeto

The screenshot displays a web application interface for managing researchers and projects. The interface is titled "Reproducibile Web" and shows the user "RODRIGO PINHEIRO".

Projeto

- Novo Projeto
- Sistema
- Sair

Listagem de pesquisadores disponíveis

(1 of 1) 10

Consultar pesquisador	Email	Site profissional	
RODRIGO PINHEIRO	rodrigo@nti.ufal.br	http://lattes.cnpq.br/0365438845525046	Associar
THAYSE DOS SANTOS FONSECA	thaysefonseca16@gmail.com	http://lattes.cnpq.br/7227645365126862	Associar
ALEJANDRO C. FRERY	acfrery@gmail.com	http://lattes.cnpq.br/6935433850568144	Associar

(1 of 1) 10

Listagem de pesquisadores associados

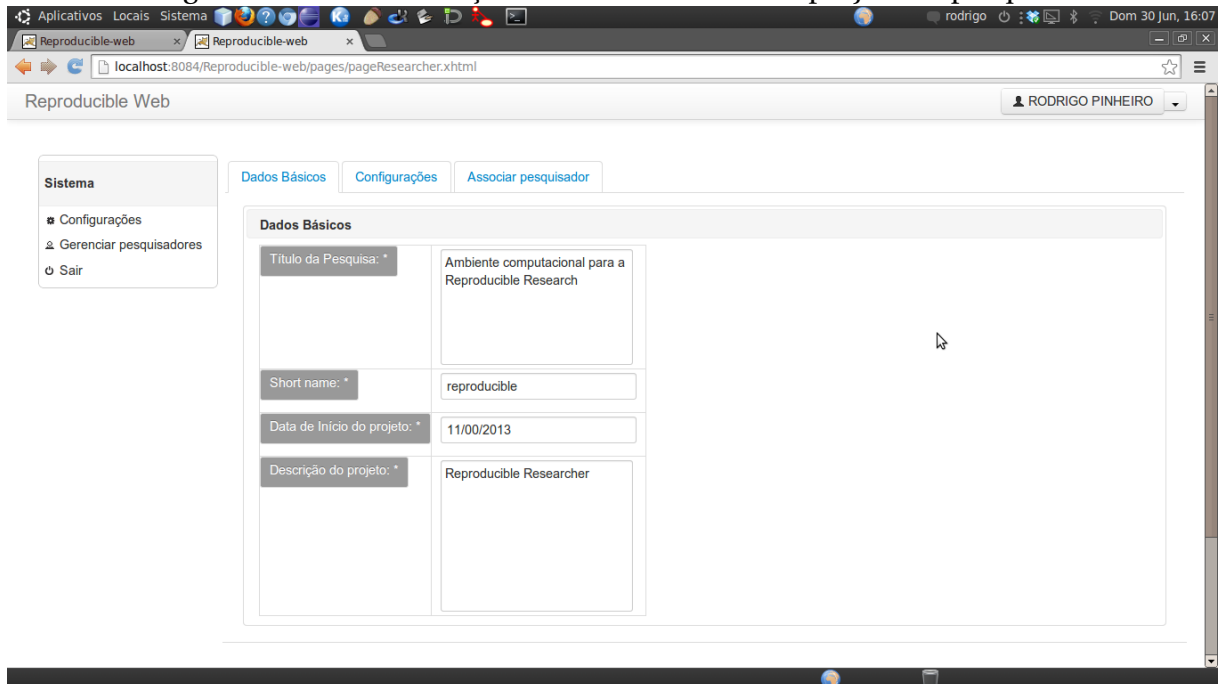
Pesquisador	Email	Site profissional	
RODRIGO PINHEIRO	rodrigo@nti.ufal.br	http://lattes.cnpq.br/0365438845525046	Remover

Salvar Limpar Home

Fonte: Autor, 2013.

A opção de editar permite a atualização dos dados cadastrais de um projeto de pesquisa. Na figura 5.11 é exibida a tela de edição dos dados cadastrais do projeto de pesquisa.

Figura 5.11: Tela de edição dos dados básicos do projeto de pesquisa

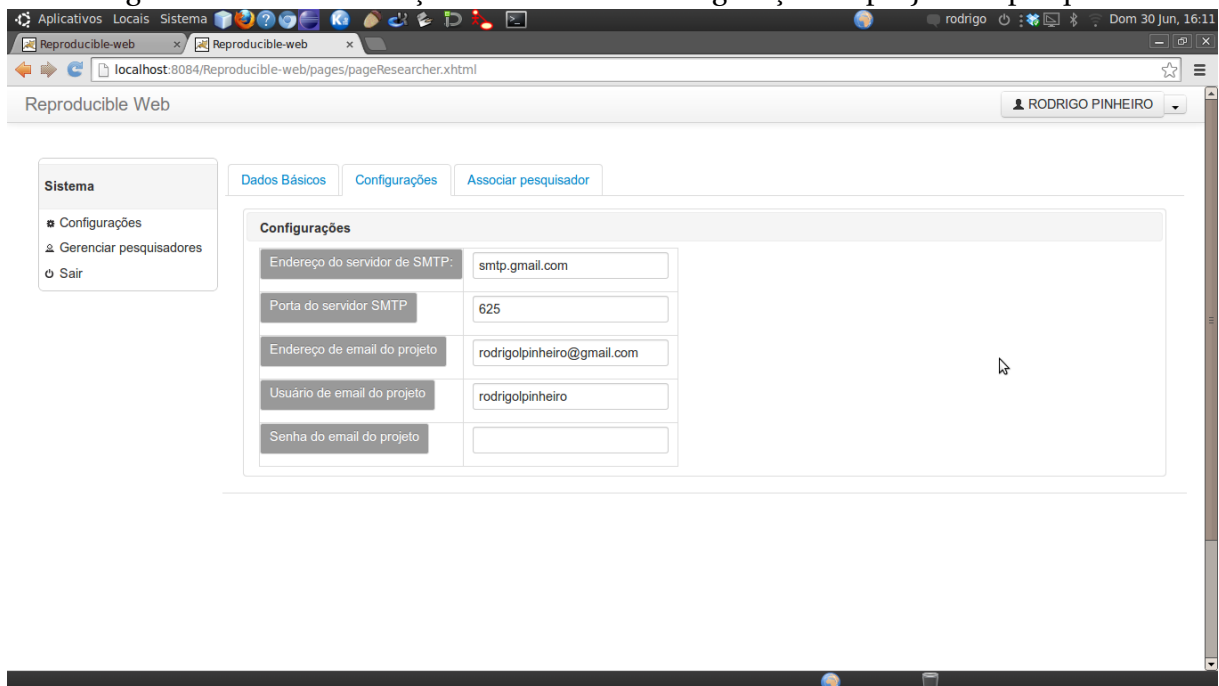


Dados Básicos	
Titulo da Pesquisa: *	Ambiente computacional para a Reproducibile Research
Short name: *	reproducibile
Data de Início do projeto: *	11/00/2013
Descrição do projeto: *	Reproducibile Reseacher

Fonte: Autor, 2013.

Na figura 5.12 é exibida a tela de edição dos dados de configuração do projeto de pesquisa.

Figura 5.12: Tela de edição dos dados de configuração do projeto de pesquisa

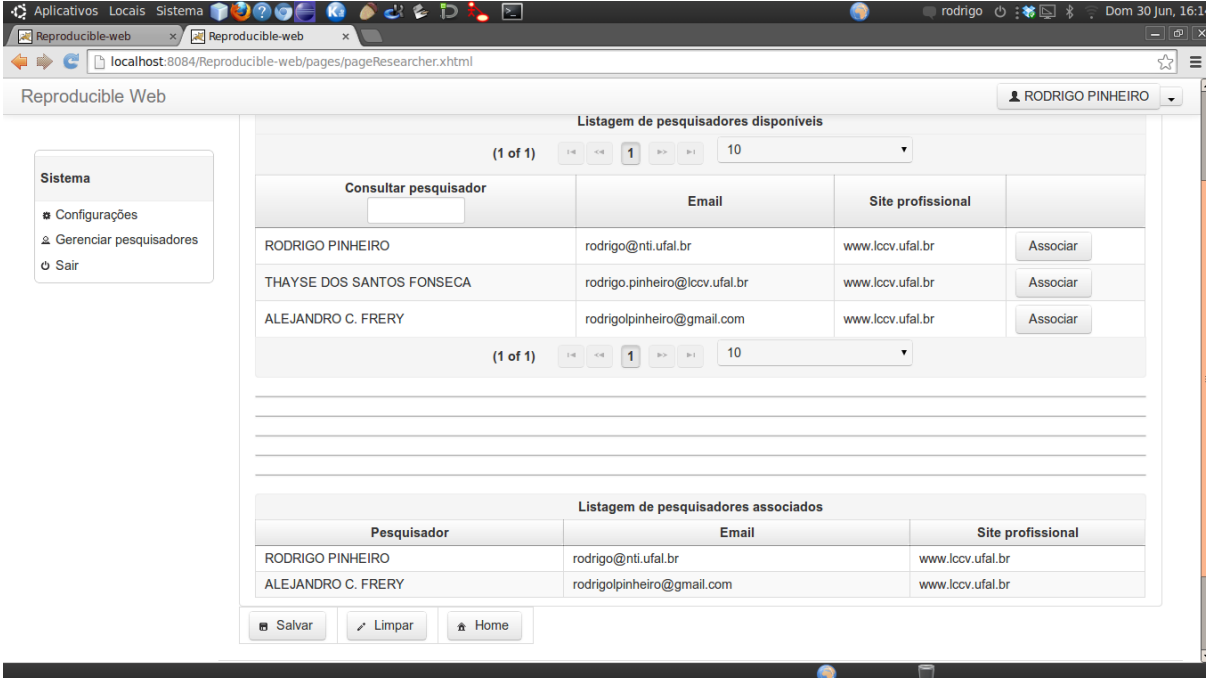


Configurações	
Endereço do servidor de SMTP:	smtp.gmail.com
Porta do servidor SMTP	625
Endereço de email do projeto	rodrigolpinheiro@gmail.com
Usuário de email do projeto	rodrigolpinheiro
Senha do email do projeto	

Fonte: Autor, 2013.

Na figura 5.13 é exibida a tela de edição da associação entre pesquisadores e um projeto de pesquisa.

Figura 5.13: Tela de edição da associação entre pesquisadores e um projeto de pesquisa



The screenshot displays a web application interface for managing researchers and projects. The interface is titled "Reproducibile Web" and shows a user profile for "RODRIGO PINHEIRO".

Sistema

- Configurações
- Gerenciar pesquisadores
- Sair

Listagem de pesquisadores disponíveis

(1 of 1) 1 10

Consultar pesquisador	Email	Site profissional	
RODRIGO PINHEIRO	rodrigo@nti.ufal.br	www.lccv.ufal.br	Associar
THAYSE DOS SANTOS FONSECA	rodrigo.pinheiro@lccv.ufal.br	www.lccv.ufal.br	Associar
ALEJANDRO C. FRERY	rodrigolpinheiro@gmail.com	www.lccv.ufal.br	Associar

(1 of 1) 1 10

Listagem de pesquisadores associados

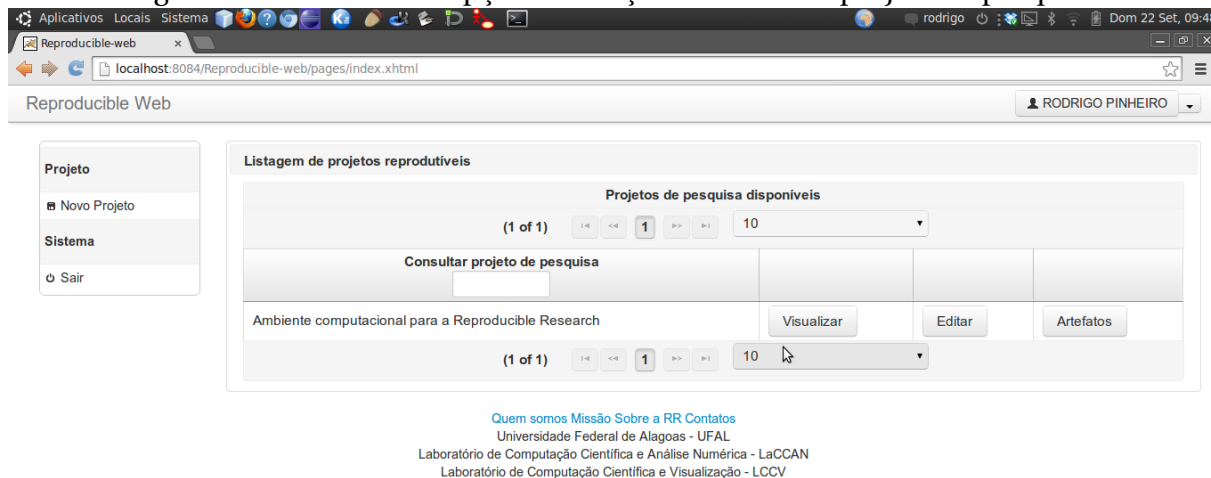
Pesquisador	Email	Site profissional
RODRIGO PINHEIRO	rodrigo@nti.ufal.br	www.lccv.ufal.br
ALEJANDRO C. FRERY	rodrigolpinheiro@gmail.com	www.lccv.ufal.br

Salvar Limpar Home

Fonte: Autor, 2013.

A opção de novo projeto permite a criação de um novo projeto de pesquisa. Na figura 5.14 percebe-se a opção de novo projeto no menu a esquerda.

Figura 5.14: Tela com a opção de criação de um novo projeto de pesquisa



Fonte: Autor, 2013.

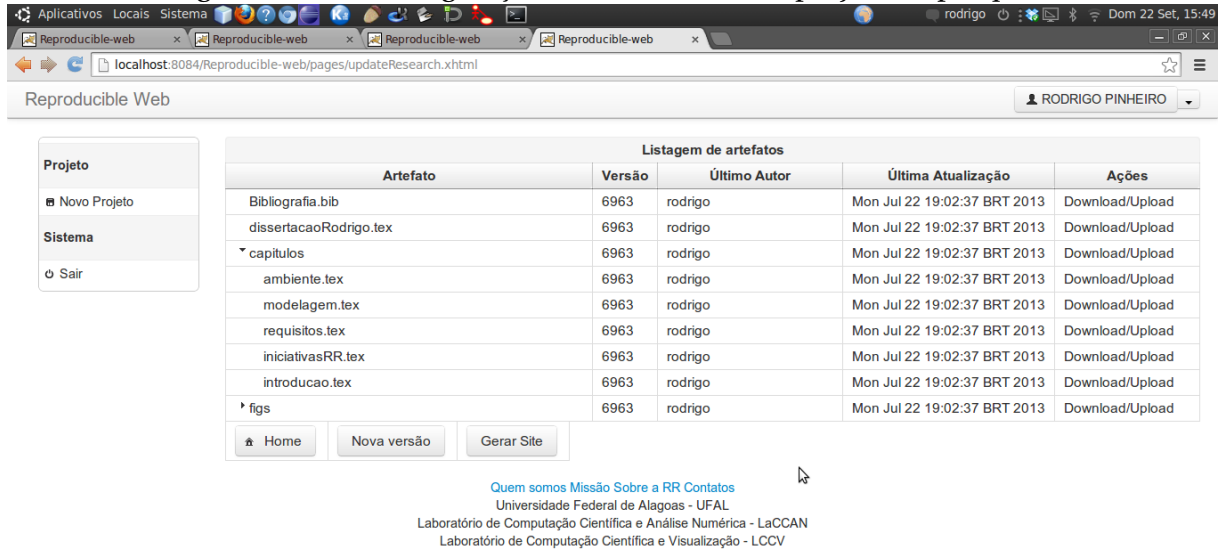
Ao utilizar a opção de novo projeto, será permitido ao pesquisador/colaborador a criação de um novo projeto de pesquisa. Essa funcionalidade utiliza as mesmas telas das opções de editar e visualizar projeto de pesquisa.

A opção de artefatos será mostrada em detalhes na seção 5.2.6.

5.2.4 Gestão de versão de um projeto de pesquisa

Essa funcionalidade refere-se à gestão (incluir) uma versão do projeto de pesquisa. Para ter acesso a essa ação é necessário possuir o papel de pesquisador.

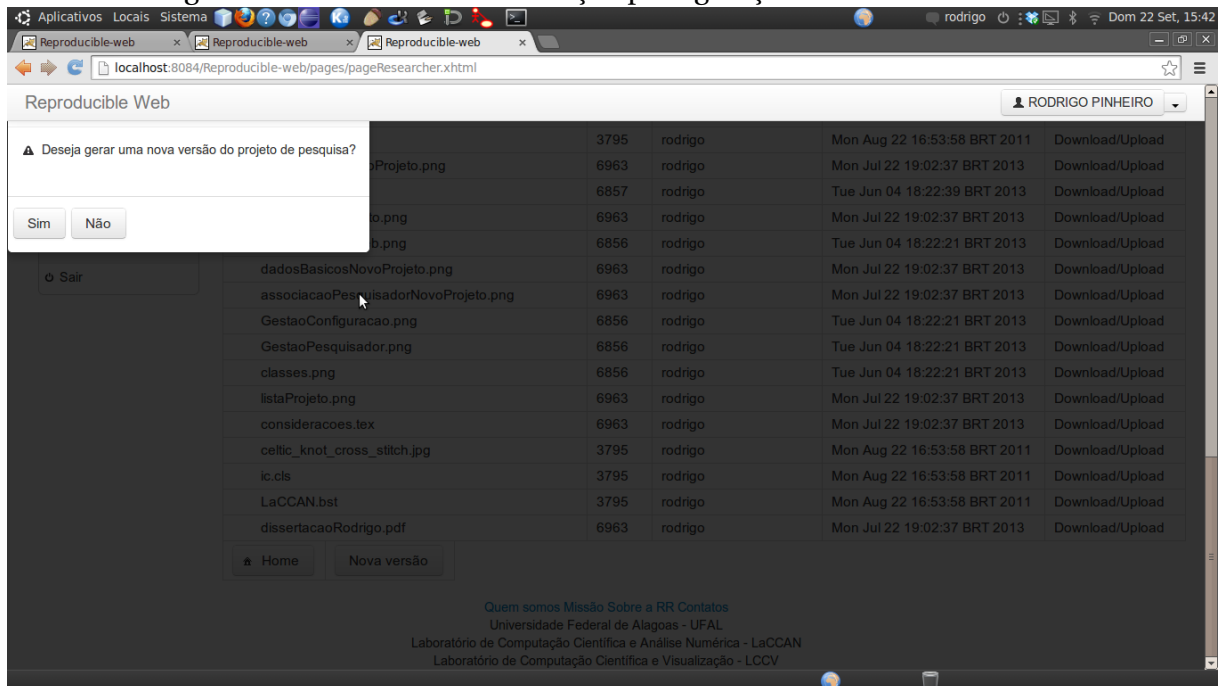
Figura 5.15: Tela de geração da nova versão do projeto de pesquisa



Fonte: Autor, 2013.

Caso o pesquisador opte por gerar uma nova versão do projeto de pesquisa, receberá a mensagem exibida na Figura 5.16.

Figura 5.16: Botão de confirmação para geração de uma nova versão

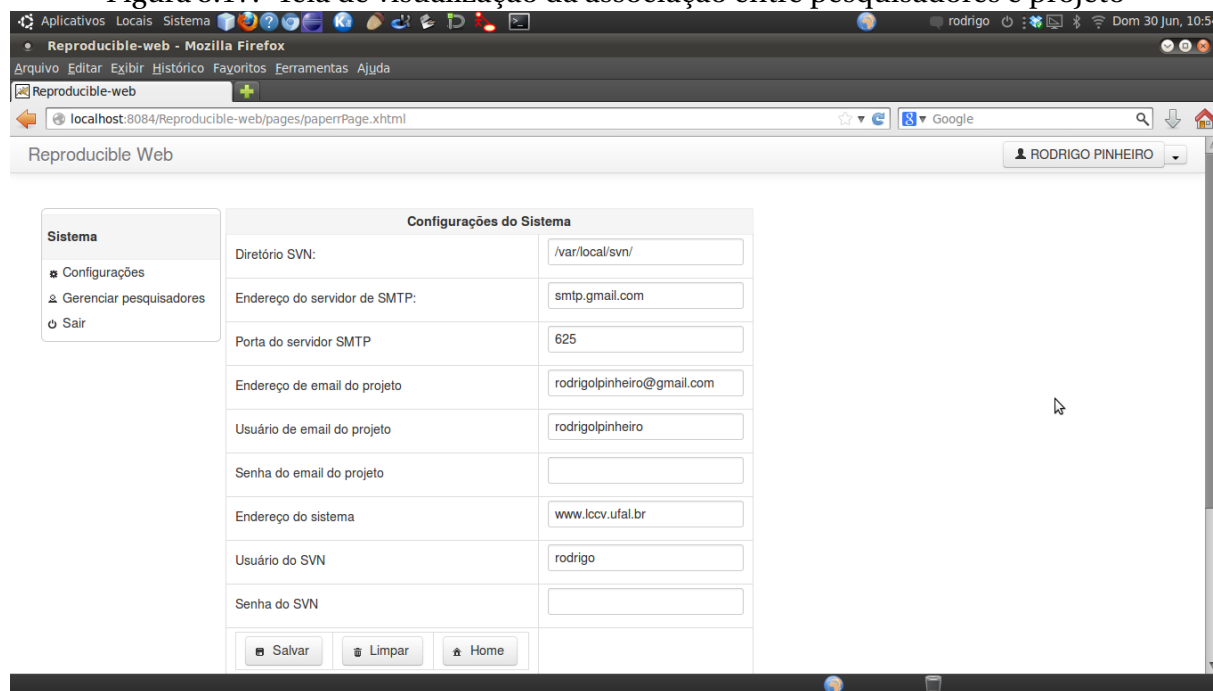


Fonte: Autor, 2013.

5.2.5 Gestão de configuração do ambiente

Essa funcionalidade refere-se à gestão (incluir e alterar) os dados de configuração do ambiente computacional. Para ter acesso a essa ação é necessário possuir o papel de administrador.

Figura 5.17: Tela de visualização da associação entre pesquisadores e projeto



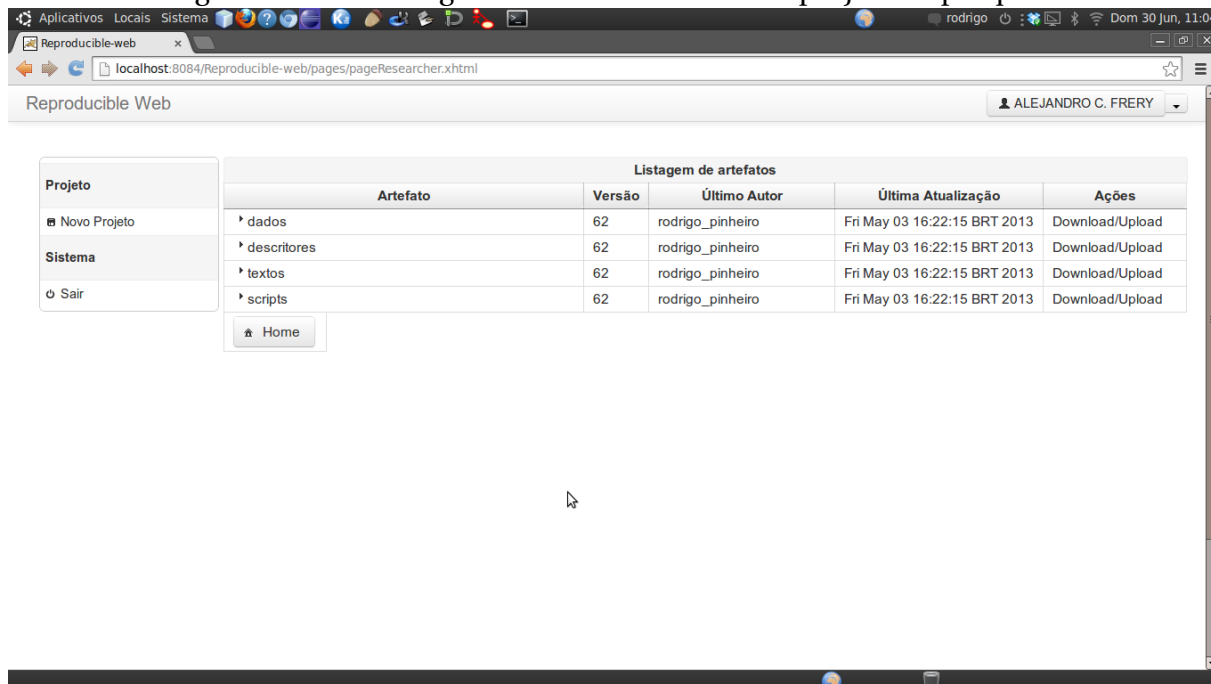
Fonte: Autor, 2013.

Entre os dados obrigatórios para o funcionamento do ambiente, estão: dados de configuração do SVN e dados de configuração de uma conta de email.

5.2.6 Gestão de artefatos do projeto de pesquisa

Essa funcionalidade refere-se à gestão (incluir, alterar, recuperar e excluir) os artefatos de um projeto de pesquisa. Para ter acesso a essa ação é necessário possuir o papel de pesquisador ou colaborador.

Figura 5.18: Tela de gestão dos artefatos de um projeto de pesquisa



Fonte: Autor, 2013.

Na Figura 5.18, o pesquisador/colaborador pode utilizar a opção de *download* para atualizar o artefato na sua máquina local e a opção de *upload* para atualizar o artefato no projeto de pesquisa.

5.2.7 Geração da página web de um projeto de pesquisa

Essa funcionalidade refere-se à gestão (incluir) uma página web para o projeto de pesquisa. Para ter acesso a essa ação é necessário possuir o papel de pesquisador.

Figura 5.19: Tela de geração da página web para o projeto de pesquisa

The screenshot shows a web browser window with the URL `localhost:8084/Reproducibile-web/pages/updateResearch.xhtml`. The page title is 'Reproducibile Web' and the user is logged in as 'RODRIGO PINHEIRO'. On the left, there is a sidebar with 'Projeto' (Novo Projeto) and 'Sistema' (Sair). The main content area is titled 'Listagem de artefatos' and contains a table with the following data:

Artefato	Versão	Último Autor	Última Atualização	Ações
Bibliografia.bib	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
dissertacaoRodrigo.tex	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
capitulos	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
ambiente.tex	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
modelagem.tex	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
requisitos.tex	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
iniciativasRR.tex	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
introducao.tex	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload
figs	6963	rodrigo	Mon Jul 22 19:02:37 BRT 2013	Download/Upload

Below the table are buttons for 'Home', 'Nova versão', and 'Gerar Site'. At the bottom, there is a footer with contact information for UFAL and LaCCAN/LCCV.

Fonte: Autor, 2013.

Caso o pesquisador opte por gerar uma página web do projeto de pesquisa, receberá a mensagem exibida na Figura

Figura 5.20: Botão de confirmação para geração de uma nova página web para o projeto de pesquisa

The screenshot shows the same 'Reproducibile Web' interface as Figure 5.19, but with a confirmation dialog box open. The dialog box has the title 'Deseja gerar um site para o projeto de pesquisa?' and two buttons: 'Sim' and 'Não'. The background content is dimmed, showing the same table of artifacts and the 'Gerar Site' button.

Fonte: Autor, 2013.

5.3 Projeto reprodutível

Para apoiar o ambiente computacional foi criado o site <https://sites.google.com/site/projetoreproducibleresearch/>.

Figura 5.21: Site de apoio ao projeto



Fonte: Autor, 2013.

Nesse site estão disponibilizadas todas as publicações que foram originadas a partir do desenvolvimento do ambiente computacional.

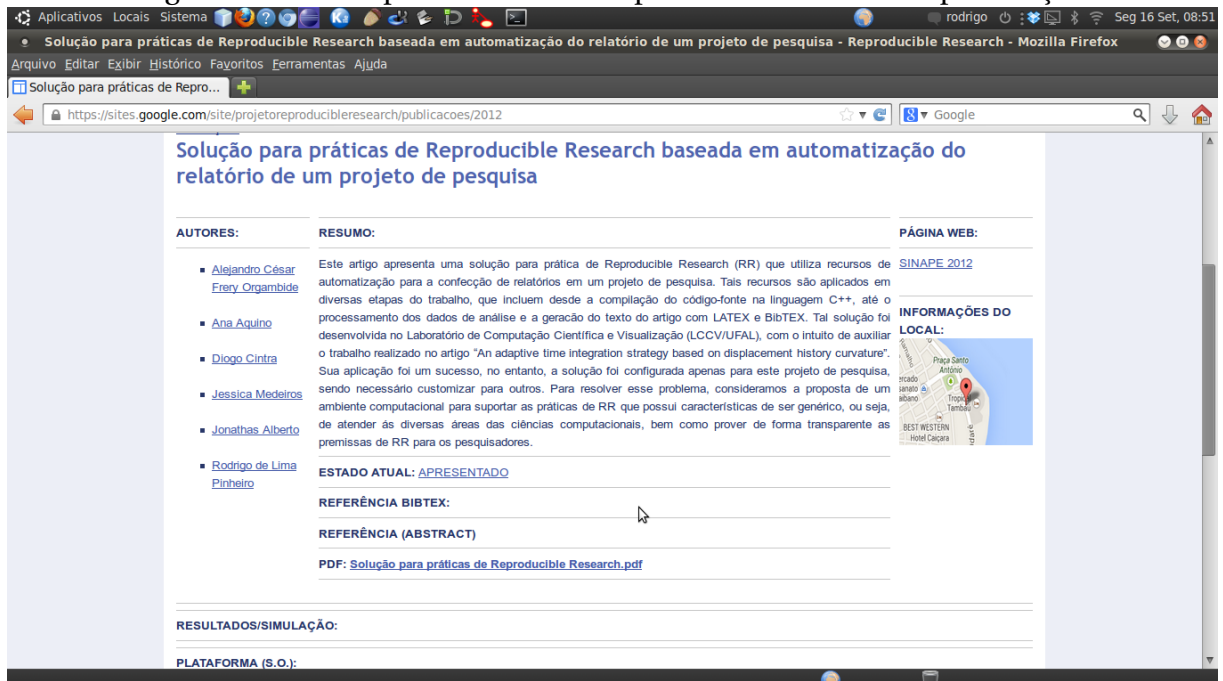
Figura 5.22: Publicações originadas a partir do desenvolvimento do ambiente computacional



Fonte: Autor, 2013.

Em cada publicação o site disponibiliza uma série de artefatos, conforme mostrado na Figura 5.23.

Figura 5.23: Exemplo de artefatos disponibilizados em cada publicação



Fonte: Autor, 2013.

Na Figura 5.24 são mostrados todos os artefatos disponibilizados para o ambiente computacional.

Figura 5.24: Exemplo de artefatos disponibilizados para o ambiente computacional



Fonte: Autor, 2013

O capítulo apresentou o ambiente computacional. No próximo capítulo serão apresentadas a conclusão e os trabalhos futuros.

6

Considerações finais

Este trabalho representa um pequeno auxílio no sentido de popularizar as práticas de RR entre a comunidade da pesquisa computacional quantitativa.

6.1 Conclusão

O propósito deste trabalho foi a concepção e o desenvolvimento de um ambiente computacional para suporte as práticas de RR. Tal ambiente encontra-se disponível em (<https://sites.google.com/site/projetoreproducibleresearch/>), de forma gratuita e sob a licença (GPL). A sua implementação deveu-se à necessidade de adequação às novas políticas adotadas por periódicos e conferências de alto nível que, cada vez mais exigem comprovação dos resultados alcançados por um artigo em pesquisa computacional quantitativa.

Como qualquer produto de software, este ambiente computacional deverá sofrer modificações (correções e novas funcionalidades), para que cada vez mais proporcione facilidades aos laboratórios, grupos e pesquisadores interessados em realizar pesquisa computacional quantitativa comprometida com as práticas de *Reproducible Research*.

O Laboratório de Computação Científica e Visualização (LCCV) e o Laboratório de Computação Científica e Análise Numérica (LaCCAN), ambos instalados na UFAL, aceitaram o desafio de implantar o ambiente computacional.

6.2 Trabalhos futuros

Os principais trabalhos futuros relacionados ao ambiente computacional são:

- Utilização de técnicas de engenharia de software para que o ambiente seja mais extensível e flexível;
- Finalizar a integração com o software de geração da página web;
- Desenvolvimento do módulo de segurança, no qual será permitido aos pesquisadores a utilização de estruturas similares a certificados digitais;
- Disponibilizar o ambiente computacional por meio de uma máquina virtual. Segundo (Howe, 2012), uma máquina virtual pode conter todo o ambiente de trabalho de um pesquisador, incluindo dados, softwares, dependências, anotações, *logs*, *scripts* e muito mais. A grande vantagem da utilização da máquina virtual reside no fato de que ela representa uma cópia completa de um sistema operacional configurado para o uso do ambiente, que pode ser instalada de maneira simples, através de um software de virtualização, no computador pessoal do pesquisador;
- Atualizar o ambiente computacional a partir das críticas, sugestões e solicitações de correção recebidas através do fórum do ambiente computacional, com o intuito de melhorar algum aspecto do ambiente computacional.

REFERÊNCIAS

- Barnes, N. (2010), 'Publish your computer code: it is good enough', *Nature* **467**, 753.
- Buckheit, J. B. & Donoho, D. L. (1995), *Wavelets and Statistics*, Springer, chapter WaveLab and Reproducible Research.
- Donoho, D. L. (2010), 'An invitation to reproducible computational research', *Biostatistics* **11**(3), 385–388.
- Donoho, D. L., Maleki, A., Rahman, I., Shahram, M. & Stodden, V. (2009), 'Reproducible Research in Computational Harmonic Analysis', *Computing in Science & Engineering* **11**(1), 8–18.
- Eaton, J. W. (2012), 'Gnu Octave and reproducible research', *Journal Of Process Control* **22**, 1433–1438.
- Eckel, S. P. & Peng, R. D. (2009), 'Interacting with local and remote data repositories using the stashR package', *Computational Statistics* **24**(2), 247–254.
- Fadili, J. M., Starck, J., Elad, M. & Donoho, D. L. (2010), 'MCALab: Reproducible Research in Signal and Image Decomposition and Inpainting', *Computing in Science & Engineering* **12**(1), 44–62.
- Fomel, S. & Claerbout, J. F. (2009), 'Reproducible Research Introduction', *Computing in Science & Engineering* **11**(1), 5–7.
- Howe, B. (2012), 'Virtual Appliances, Cloud Computing, and Reproducible Research', *Computing in Science & Engineering* **14**(4), 36–41.
- Ince, D., Hatton, L. & Graham-Cumming, J. (2012), 'The case for open computer programs', *Nature* **482**(7386), 485–488. URL <http://www.nature.com/nature/journal/v482/n7386/full/nature10836.html>.
- Leisch, F. (2002), Sweave: Dynamic generation of statistical reports using literate data analysis, in 'Compstat 2002 - Proceedings in Computational Statistics'.
- Lenth, R. & Hojsgaard, S. (2007), 'SASweave: Literate Programming Using SAS', *Journal of Statistical Software* **19**(8), 1–20. URL <http://www.jstatsoft.org/v19/i08>.
- LeVeque, R. J., Mitchell, I. M. & Stodden, V. (2012), 'Reproducible Research for Scientific Computing: Tools and Strategies for Changing the Culture', *Computing in Science & Engineering* pp. 13–17.

- Li-Thiao-Te, S. (2012), Literate Program Execution for Reproducible Research and Executable Papers, *in* 'PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE', Vol. 9, pp. 439–448.
- Morin, A., Urban, J., Adams, P. D., Foster, I., Baker, D. & Sliz, P. (2012), 'Shining Light in Black Boxes', *Science* **336**, 159–160.
- Paula Filho, W. P. (2010), *Engenharia de Software Fundamentos, Métodos e Padrões*, 3 ed., LTC.
- Peng, R. D. (2011), 'Reproducible Research in Computational Science', *Science* **334**, 1226–1227. URL <http://www.sciencemag.org/content/334/6060/1226>.
- Schulte, E., Davison, D., Dye, T. & Dominik, C. (2012), 'A Multi-Language Computing Environment for Literate Programming and Reproducible Research', *Journal of Statistical Software* **46**, 1–24. URL <http://www.jstatsoft.org/v46/i03>.
- Schwab, M., Karrenbach, N. & Claerbout, J. (2000), 'Making scientific computations reproducible', *Computing in Science & Engineering* **2**(6), 61–67.
- Silberschatz, A., Korth, H. F. & Sudarshan, S. (1999), *Sistema de banco de dados*, 3 ed., Pearson Education do Brasil.
- Stodden, V. (2009), 'The Legal Framework for Reproducible Scientific Research: Licensing and Copyright', *Computing in Science & Engineering* **11**(1), 35–40.
- Stodden, V., Donoho, D. L. & Fomel, S. (2010), 'Reproducible Research addressing the need for data and code sharing in computational science', *Computing in Science & Engineering* **12**(5), 8–13.

Este trabalho foi redigido em \LaTeX utilizando uma modificação do estilo IC-UFAL. As referências bibliográficas foram preparadas no JabRef e administradas pelo $\text{BIB}\text{\LaTeX}$ com o estilo LaCCAN. O texto utiliza fonte Fourier-GUTenberg e os elementos matemáticos a família tipográfica Euler Virtual Math, ambas em corpo de 12 pontos. A numeração dos capítulos segue com a família tipográfica Art Nouveau Caps.

