UFRRJ

INSTITUTO DE CIÊNCIA EXATAS

CURSO DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

DISSERTAÇÃO

RFlow: uma arquitetura para execução e coleta de proveniência de *workflows* estatísticos

José Antônio Pires do Nascimento

2015

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE CIÊNCIAS EXATAS
CURSO DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

RFLOW: UMA ARQUITETURA PARA EXECUÇÃO E COLETA DE PROVENIÊNCIA DE WORKFLOWS ESTATÍSTICOS

JOSÉ ANTÔNIO PIRES DO NASCIMENTO

Sob a Orientação do Professor

Sérgio Manuel Serra da Cruz

e Coorientação do Professor

Marcos Baccis Cedia

Dissertação submetida como requisito parcial para obtenção do grau de **Mestre em Ciências,** no Curso de Pós-Graduação em Modelagem Matemática e Computacional, Área de Concentração em Inteligência Computacional e Otimização

SEROPÉDICA – RJ
2015

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
DEPARTAMENTO DE MATEMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

JOSÉ ANTÔNIO PIRES DO NASCIMENTO

Dissertação submetida como requisito parcial para obtenção do grau de **Mestre em Ciências**, no Curso de Pós-Graduação em Modelagem Matemática e Computacional, área de Concentração em Inteligência Computacional e Otimização.

DISSERTAÇÃO APROVADA EM 07/10/2015

_____
Sérgio Manuel Serra da Cruz. Dr., UFRRJ
(Orientador)

_____
Guilherme Montandon Chaer. Dr., Embrapa

_____
Raimundo José Macário Costa. Dr., UFRRJ

DEDICATÓRIA

A Deus por tudo, a meu Pai José Bernardes do Nascimento (in memorian) e à minha Mãe Perciliana Pires do Nascimento pelos exemplos de fé, a meus Irmãos pelo apoio, à Dulce, Matheus e Natália pelo carinho e paciência que tiveram comigo durante o curso.

"Entrega o teu caminho ao Senhor; confia Nele, e o mais Ele fará" Salmo 37:5

AGRADECIMENTOS

Primeiramente a Deus e seu Filho unigênito Jesus Cristo, criador de todas as coisas, pelo dom da vida e todas as graças concedidas.

À minha família que sempre esteve presente nos momentos bons e ruins, me apoiando e compartilhando minhas alegrias e tristezas.

Ao meu orientador Sérgio Manuel Serra da Cruz, por ter acreditado no meu trabalho desde o início e revisado os textos com maestria.

À pesquisadora Janaina Rouws, minha conselheira acadêmica na Embrapa Agrobiologia, pelos conhecimentos transmitidos e estar presente quando precisava.

Aos professores Guilherme Montandon Chaer e Raimundo José Macário Costa pela participação na banca examinadora e as observações pertinentes na dissertação.

A todos os professores do PPG-MMC, principalmente os que me deram aula, não mediram esforços para compartilhar o conhecimento.

Aos colegas do NTI da Embrapa Agrobiologia: Fernando, Hugo e Jayme, pelo total apoio. O Hugo pela ajuda na infraestrutura, o Fernando e o Jayme pelos momentos de descontração.

À secretária Janaina Gama do Departamento de Matemática, atende aos alunos com presteza e simpatia.

Aos colegas de turma, principalmente ao Tárcio Triani pelos ensinamentos de equações diferenciais e à Kelly Harumi pelas boas conversas.

À Embrapa e seus gestores, por estimularem seus colaboradores a se capacitarem.

À UFRRJ, pela excelência no ensino.

RESUMO

NASCIMENTO, José Antônio Pires do. **RFlow: uma arquitetura para execução e coleta de proveniência de workflows estatísticos. 2015**. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Programa de Pós-Graduação em Modelagem Matemática e Computacional, Departamento de Matemática, Universidade Federal Rural do Rio de Janeiro, Seropédica, 2015.

Os dados agropecuários relacionados à redução de custos de produção e aumento da qualidade de produtos, previsão e controle de pragas e epidemias e agricultura de alta precisão são produzidos em grande escala e de maneira heterogênea e distribuída através de sensores, VANTs, web, satélites, dispositivos móveis, planilhas, entre outros. Este crescente aumento no volume de dados científicos e a necessidade de gerenciá-los e compartilhá-los entre equipes geograficamente dispersas têm demandado novas técnicas e ferramentas computacionais. Este trabalho apresenta a arquitetura RFlow, um conjunto de ferramentas integradas, com o intuito de gerenciar, compartilhar e reproduzir os experimentos científicos baseados em *scripts* R legados e, também, auxiliar a validar os resultados estatísticos junto à comunidade científica. O aplicativo SisGExp, um dos componentes da arquitetura, permite não só o acesso aos dados e os processos que os transformaram via *online*, bem como a coleta e registro dos descritores de proveniência sobre os experimentos. Além disso, vincula os dados de pesquisa aos resultados estatísticos, o que amplia a reprodutibilidade do experimento, oferecendo maior confiabilidade aos resultados científicos.

PALAVRAS-CHAVE: *Workflow* Científico, Proveniência, Sistema R, Agropecuária.

ABSTRACT

NASCIMENTO, José Antônio Pires do. **RFlow: uma arquitetura para execução e coleta de proveniência de workflows estatísticos. 2015**. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Programa de Pós-Graduação em Modelagem Matemática e Computacional, Departamento de Matemática, Universidade Federal Rural do Rio de Janeiro, Seropédica, 2015.

Agricultural data related to the reduction of costs of production and improvements in product quality, prediction and control of pests and epidemics and high precision agriculture are produced on a large scale, and in a heterogeneous manner. The data are captured via sensors, UAVs, the web, satellites, mobile devices, among others. This increasing volume of scientific data and the need to manage and share them between geographically dispersed teams has created a demand for new techniques and computational tools. This study presents the RFlow architecture, a set of integrated tools that manages, shares and collects provenance and reproduces scientific experiments based on R scripts and helps to validate the statistical results. The SisGExp application, one of the architectural components, allows not only access to the data and the processes that are transformed in real time but also collects and records prospective and retrospective provenance descriptors concerning the experiment. In addition, the alignment of the research data to statistical results expands experimental reproducibility, providing greater reliability of scientific results.

KEYWORDS: Scientific workflow, Provenance, R System, Agriculture.

LISTA DE QUADROS

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Poder reproduzir e compartilhar os experimentos e seus respectivos resultados de forma confiável, imediata e a qualquer tempo e lugar é um dos desafios na pesquisa científica. Não menos importante na pesquisa agropecuária, que vem produzindo dados em grande escala e de maneira heterogênea e distribuída através de sensores, Veículos Aéreos Não Tripulados (VANTs), web, satélites, bancos de dados distribuídos e dispositivos móveis. Grande parte desses dados são processados em *scripts* legados desenvolvidos na linguagem R (R DEVELOPMENT CORE TEAM, 2012). Isso motivou a concepção e criação da arquitetura RFlow (NASCIMENTO; CRUZ, 2013), que visa o gerenciamento dos *workflows* estatísticos (baseados em *scripts* R) para mitigar algumas das limitações dos *softwares* estatísticos no que diz respeito ao gerenciamento da proveniência prospectiva e retrospectiva (MAIR; DE LEEUW, 2010).

Nesta introdução são destacadas a contextualização da pesquisa e a motivação para o desenvolvimento da arquitetura RFlow. São definidos o problema de pesquisa, a hipótese e os objetivos que nortearão este trabalho. Além disso, haverá a delimitação e o escopo do problema tratado pela arquitetura e uma visão resumida do planejamento inicial para alcançar os objetivos propostos.

1.1 Contexto Geral

Crescimento populacional, mudanças climáticas, bioenergia são problemas de escala global que estão demandando a integração de diversas ciências para melhorar as práticas agrícolas. Acrescente-se a isso a necessidade de aprofundar as pesquisas nas áreas de degradação do solo, perda de biodiversidade e desaceleração do crescimento na produtividade das culturas (TUOT et al., 2008). A junção dessas áreas aliadas aos avanços na área de processamento de alto desempenho (PAD) e *e-Science* (GRAY, 2009), da rápida produção de grandes volumes de dados científicos na Agricultura através de sensores, satélites e modernos aparatos experimentais (BRANCH et al., 2014) e da urgente necessidade da garantia de reprodutibilidade dos experimentos têm exigido novas abordagens no que tange à gerência de grandes volumes de dados e dos descritores de proveniência em experimentos científicos distribuídos do tipo *in silico*. Tudo isso tem contribuído para o aumento sistemático do uso de *workflows* científicos (DEELMAN et al., 2009, MATTOSO et al., 2009).

Workflow científico pode ser compreendido como o encadeamento de atividades no contexto de um experimento científico que manipula dados com o intuito de atingir um determinado resultado (MATTOSO et al., 2009). Os *workflows* científicos passaram a ser incorporados em projetos de *e-Science*. Eles são utilizados para representar abstrações sobre experimentos realizados no computador, permitindo uma composição estruturada de programas sob a forma de sequência de atividades. Para automatizar a construção e execução dos *workflows* científicos foram desenvolvidas ferramentas computacionais denominadas sistemas de gerência de *workflows* científicos (SGWfC) (TAYLOR et al*.,* 2007), (HEY et al., 2009). Os SGWfC possibilitam os pesquisadores realizarem uma espécie de programação em alto nível através do encadeamento de processos científicos (ou atividades) que seguem uma determinada lógica.

A utilização de *workflows* científicos com foco no processamento estatístico também vem crescendo (NASCIMENTO; CRUZ, 2013). Os *workflows* estatísticos se caracterizam pela capacidade de manipular grandes volumes de dados e por executarem sofisticadas análises estatísticas através da incorporação de recursos (funções, algoritmos e métodos) disponíveis em sistemas estatísticos tradicionais (SPSS, SAS, Statistica, Mapple, MathLab, Weka, R, entre outros (MAIR; DE LEEUW, 2010)). Os sistemas estatísticos possuem características peculiares: podem ser de código aberto ou proprietários ou disponibilizarem recursos estatísticos e gráficos com distintos graus de sofisticação e precisão. No entanto, como característica comum, sua utilização requer sólidos conhecimentos em Estatística por parte dos usuários.

O Sistema R é um dos *softwares* estatísticos mais difundidos na atualidade, sendo amplamente utilizado tanto nas áreas comerciais e científicas, como em diversas empresas, universidades e institutos de pesquisa. Ele é capaz de executar desde simples comandos *online* até longos e sofisticados *scripts* desenvolvidos de modo *ad hoc* (CRAWLEY, 2002). No entanto, ele apresenta uma limitação: ainda não dispõe de recursos de captura de proveniência sobre os processamentos e análises estatísticas realizadas pelos pesquisadores (RUNNALLS, 2013). Coletar proveniência nos *scripts* é um desafio que ainda se encontra em aberto (NASCIMENTO; CRUZ, 2013), (MURTA et al, 2014).

Este trabalho apresenta a abordagem denominada RFlow, que facilita o gerenciamento de experimentos apoiados por *workflows* estatísticos baseados em *scripts* R. Visa mitigar limitações dos sistemas estatísticos no que diz respeito à coleta transparente e não intrusiva de proveniência prospectiva e retrospectiva (BUNEMAN et al*.,* 2001). A arquitetura proposta é multimodular e representa uma concepção que permite os pesquisadores (re)utilizarem os *scripts* R encapsulados sob a forma de meta-*workflows* científicos, facilitando o reúso de dados e dos próprios *scripts* R. Além disso, permite o compartilhamento e o acompanhamento de cada atividade dos experimentos com apoio da coleta de descritores de proveniência sobre as execuções individualizadas de cada instância do *workflow.*

1.2 Contexto na Embrapa

A Empresa Brasileira de Pesquisa Agropecuária (Embrapa) está vinculada ao Ministério da Agricultura, Pecuária e Abastecimento (Mapa), foi criada em abril de 1973 e atualmente possui 17 Unidades centrais localizadas em Brasília, 46 Unidades descentralizadas instaladas em todo o Brasil, além de escritórios e laboratórios no exterior. Desde a sua criação, assumiu como desafio desenvolver em conjunto com parceiros do Sistema Nacional de Pesquisa Agropecuária (SNPA), um modelo de agricultura e pecuária tropical genuinamente brasileiro, superando as barreiras que limitavam a produção de alimentos, fibras e energia no Brasil (EMBRAPA, 2015).

Esse esforço ajudou a transformar o Brasil. Hoje a agropecuária brasileira é uma das mais eficientes e sustentáveis do planeta. Incorporou uma larga área de terras degradadas dos cerrados aos sistemas produtivos. Permitiu a quadruplicação da oferta de carne bovina e suína e ampliou em 22 vezes a oferta de frango. Essas são algumas das principais conquistas que tiraram o Brasil de uma condição de importador de alimentos básicos para a condição de um dos maiores produtores e exportadores mundiais (EMBRAPA, 2015).

Dessa maneira, para alcançar tais resultados muitas pesquisas são diariamente desenvolvidas em suas Unidades. Atualmente na Embrapa, milhares de arquivos de dados e

scripts são constantemente gerados em experimentos ou unidades observacionais. Muitos desses arquivos de dados pertencem a trabalhos desenvolvidos por bolsistas de iniciação científica, mestrado e doutorado, juntamente aos pesquisadores das Unidades. Ocorre que muitos desses arquivos e dados, que pertencem à Embrapa, são perdidos ao longo do tempo com a saída do bolsista da Unidade ou muitas vezes por falta de uma maior padronização e/ou sistematização na coleta e registro dos dados (Comunicação pessoal (Janaina Ribeiro Costa Rouws, 2014, Embrapa Agrobiologia)).

Nesse sentido, procedimentos de análises estatísticas tornam-se limitados ou mesmo quase impossíveis de serem realizados se as informações necessárias a respeito dos dados estão insuficientes, e isso pode comprometer toda a interpretação e conclusão dos resultados (Comunicação pessoal (Janaina Ribeiro Costa Rouws, 2014, Embrapa Agrobiologia).

Uma outra dificuldade é que muitas vezes não há uma interação maior entre o pesquisador e o profissional especializado da área de Estatística para o planejamento do experimento e posterior análises de seus dados. Algumas vezes é o bolsista quem fica encarregado dessa parte, sem ter ainda um conhecimento suficiente sobre o que se está pesquisando ou mesmo sobre todo o procedimento experimental, tornando difícil a comunicação com o profissional da área de estatística. Isso pode prejudicar o planejamento do experimento, afetar as análises e a interpretação dos resultados (Comunicação pessoal (Janaina Ribeiro Costa Rouws, 2014, Embrapa Agrobiologia)).

Portanto, verifica-se que a ausência de padronização e sistematização no planejamento de um experimento poderá incorrer em mau uso dos recursos, pois elevará o custo, tempo e esforço humano para reparar o erro. Associado a tudo isso está a impossibilidade ou dificuldade de replicação da pesquisa, uma vez que a proveniência dos dados e processos estão incompletos ou em casos mais graves estão ausentes.

1.3 Problema de Pesquisa

A definição de um problema de pesquisa é uma etapa fundamental de qualquer investigação científica. Ela consiste em expor de modo claro, compreensível e operacional qual a dificuldade com a qual nos defrontamos e que pretendemos resolver (LAKATOS; MARCONI, 1991).

O problema avaliado nesta dissertação de mestrado tem como base a aplicação de conhecimentos da Ciência da Computação, em especial Banco de Dados, modelagem computacional e desenvolvimento de sistemas no domínio da *e-Science*. Especificamente, são investigadas as questões inerentes à gestão de experimentos e de grandes volumes de dados científicos coletados ao longo do tempo por equipes distintas.

O problema da pesquisa pode ser enunciado da seguinte forma:

> "Como gerenciar, recuperar e reproduzir a qualquer momento e em qualquer Unidade da Embrapa: dados, análises estatísticas e os resultados experimentais baseados em *scripts* legados R referentes a um experimento *y* realizado pelo pesquisador *x* na data *d1*?"

1.4 Hipótese de Pesquisa

A hipótese de pesquisa é um elemento fundamental de qualquer pesquisa científica. Ela é uma proposição antecipatória à comprovação de uma realidade existencial. É um tipo de pressuposição que antecede a constatação dos fatos científicos (BARROS; LEHFELD, 1999).

Uma hipótese de pesquisa deve estar fundamentada até certo ponto em conhecimentos anteriores (LAKATOS; MARCONI, 1991). Ela deve ser compatível com o corpo do conhecimento científico já existente e pode ser testada e avaliada como provavelmente verdadeira ou falsa.

A hipótese desta pesquisa foi enunciada da seguinte forma:

> "Acredita-se que a utilização de uma instância da arquitetura RFlow (NASCIMENTO; CRUZ, 2013) permite a reprodutibilidade dos experimentos científicos baseados em *workflows* estatísticos e o gerenciamento de dados e resultados a qualquer momento, em lugares distintos e com baixo esforço por parte dos pesquisadores."

1.5 Objetivos

Esta seção apresenta os objetivos gerais e específicos desta pesquisa.

1.5.1 Objetivo Geral

O objetivo desse trabalho é apresentar uma arquitetura denominada RFlow, que permite o gerenciamento dos *workflows* estatísticos e mitigar parte das limitações dos sistemas estatísticos no que diz respeito à ausência de gerenciamento da proveniência. A arquitetura RFlow encapsula *scripts* legados descritos em linguagem R sob a forma de meta-*workflows,* que proporcionam o reúso, compartilhamento e controle de execução com apoio da coleta de proveniência sobre as execuções individualizadas de *scripts* R. O conceito meta-*workflow* indica uma abstração. Entretanto, ele pode se materializar sob a forma de um *workflow* concreto passível de ser executado por um sistema gerenciador de *workflows* científicos (SGWfC) (KUMAR; WAINER, 2005).

A arquitetura RFlow auxilia aqueles pesquisadores que: (i) utilizam *scripts* R legados como parte dos seus experimentos científicos; (ii) possuem limitados conhecimentos sobre como desenvolver novos *workflows* científicos ou não dispõem de recursos nem tempo para refatorar *scripts* preexistentes no R e; (iii) necessitam dos descritores de proveniência coletados durante a execução do experimento para gerar os resultados estatísticos *online*.

A arquitetura não requer recodificação ou alteração de códigos-fonte dos *scripts* R legados, ela reutiliza recursos do SGWfC e do meta-*workflow* para possibilitar a coleta de descritores de proveniência produzidos durante a execução de *scripts* R.

1.5.2 Objetivos Específicos

1. Padronizar através do aplicativo SisGExp (NASCIMENTO; CRUZ, 2015) a coleta dos dados gerados em experimentos e/ou estudos observacionais de pesquisadores,

analistas e bolsistas da Embrapa Agrobiologia, e assim facilitar a comunicação entre os atores envolvidos no processo.

2. Atuar como repositório para os *scripts* R, arquivos de dados e as publicações científicas relacionadas aos experimentos previamente cadastrados.
3. Permitir através do perfil do usuário no SisGExp o acompanhamento de cada etapa do processo de experimentação de maneira individualizada.
4. Gerar resultados estatísticos *online* através dos *scripts* R previamente cadastrados, bem como recuperar possíveis erros na execução do *script* R.

1.6 Área de Pesquisa

Apesar desta dissertação ter um viés multidisciplinar está fortemente baseada na Ciência da Computação, pois perpassa a área de banco de dados quando se utiliza modelagem de dados, proveniência e Sistema Gerenciador de Banco de Dados (SGBD) relacionais para registrar os experimentos, seus dados e metadados; a engenharia de software devido ao desenvolvimento do SisGExp para a coleta e visualização dos dados provenientes de experimentos e/ou unidades observacionais e a modelagem subjacente com a utilização do sistema estatístico R e do SGWfC Kepler (ALTINTAS et al., 2004) para a execução de *workflows* estatísticos (baseados em *scripts* R). Além disso, a arquitetura RFlow pode ser estendida para vários domínios da ciência: biologia, matemática, bioinformática, estatística, informática, entre outros.

1.7 Planejamento do Estudo

Para se alcançar os objetivos propostos por esta dissertação foram seguidos os seguintes passos: (i) conhecer a área de experimentação agrícola, desde o planejamento do experimento até as análises dos resultados estatísticos. Para isso, foram realizadas entrevistas na Embrapa Agrobiologia com Pesquisadores e Bolsistas para entender o processo de instalação e acompanhamento de experimentos agrícola; (ii) foram realizadas entrevistas com a profissional de estatística para obter informações sobre planejamento e análises estatísticas; (iii) por fim, foram feitas revisões da literatura sobre experimentação agrícola, estatística, *workflow* científico, SGWfC, e-*Science*, proveniência, entre outros.

O sistema R possui destaque nesse estudo, pois além de ser de livre acesso é um dos softwares estatísticos mais utilizados na Embrapa para realizar análises estatísticas. Porém, há a necessidade de integrar os experimentos agropecuários com os resultados estatísticos gerados pelo sistema R, desse modo será possível a validação do resultado estatístico e ainda a reprodução do experimento. A partir dessa necessidade iniciou-se estudos e testes com SGWfC, uma vez que alguns desses softwares possuem toda a infraestrutura (banco de dados, sistema R, proveniência) capaz de executar *scripts* R e gerar proveniência retrospectiva.

A arquitetura RFlow (NASCIMENTO; CRUZ, 2013) foi concebida para atender a lacuna existente entre *workflows* estatísticos e a coleta de proveniência. No início, RFlow era composta das ferramentas SGWfC Kepler, Sistema R, *workflow* científico ExecScript e o banco de dados PostgreSQL. Gerava apenas proveniência retrospectiva. Para gerar proveniência prospectiva e resultados estatísticos *online* foi concebido e desenvolvido o

SisGExp (NASCIMENTO; CRUZ, 2015). O aplicativo está integrado na arquitetura RFlow e foi desenvolvido na linguagem Java para a Web.

O trabalho adota como estratégia de pesquisa o estudo de caso e o método de pesquisa empregado é o qualitativo. A avaliação da arquitetura RFlow é feita através de estudos de caso, o que permite verificar o funcionamento dos componentes da arquitetura, principalmente o módulo coletor de proveniência (ator *Provenance Recorder* (ALTINTAS et al., 2006)), componente do SGWfC Kepler responsável pela proveniência retrospectiva. A vinculação da proveniência prospectiva com a proveniência retrospectiva é realizada no momento que o SisGExp invoca o Kepler para processar o *script* R legado.

São utilizados *scripts* R de domínio público baixados da Internet, bem como *scripts* da própria Embrapa (são utilizados dados artificiais devido ao termo de confidencialidade e ética da Embrapa). O uso de *scripts* de domínio público se deve a possibilidade de avaliar a capacidade da arquitetura RFlow de processar outras categorias de *scripts* R que não sejam apenas os *scripts* ligados aos experimentos agropecuários.

1.8 Estrutura da Dissertação

Esta dissertação está organizada em oito capítulos, incluída a Introdução. O Capítulo 2 apresenta uma breve definição dos principais conceitos que fundamentam a arquitetura RFlow, como: experimentação na área agrícola, ciclo de vida de experimentos científicos, *e-Science*, *workflows* científicos, *workflows* estatísticos, sistema R, SGWfC e proveniência de dados, além de citar alguns trabalhos relacionados à proveniência e execução de *scripts* R. O Capítulo 3 apresenta os materiais e métodos utilizados para implementar a arquitetura RFlow e seus componentes, bem como os experimentos utilizados para avaliá-la. O Capítulo 4 dá uma visão geral da arquitetura RFlow e suas camadas. O Capítulo 5 apresenta o aplicativo SisGExp e faz uma correlação com os experimentos agrícolas, desde o planejamento do experimento até os resultados estatísticos. O Capítulo 6 avalia a arquitetura através de três estudos de caso e discute os resultados obtidos. O Capítulo 7 apresenta a conclusão da dissertação, as principais contribuições e limitações da mesma, além de relacionar possíveis trabalhos futuros que podem ser desenvolvidos. Por fim, o Capítulo 8 apresenta as referências utilizadas nesta dissertação.

2 REFERENCIAL TEÓRICO

Neste capítulo são abordados os principais conceitos no planejamento de um experimento agrícola, uma vez que esse conjunto de conhecimentos (desde a formulação da hipótese até a análise dos resultados) são utilizados pela arquitetura proposta. Além disso, são apresentados os principais conceitos relacionados com a modelagem computacional baseada em *e-Science*, *Workflows* Científicos e proveniência. Por fim, são avaliados e relacionados os principais trabalhos na área, sendo apresentados o diferencial de cada um e um comparativo com a arquitetura RFlow.

2.1 Experimentação Agrícola

Ronald A. Fisher foi o matemático inglês que entre 1919 a 1925 formulou os princípios básicos da experimentação enquanto trabalhava na Estação Experimental de Agricultura de Rothamstead na Inglaterra. Ele desenvolveu a teoria e os métodos da pesquisa agrícola que se tornaram a base da estatística moderna. Esses métodos estatísticos são aplicados a outras áreas da ciência e da tecnologia, sendo mantidos alguns termos da pesquisa agrícola como tratamentos, parcelas, dentre outros na aplicação da estatística experimental (BANZATTO; KRONKA,1992).

O objetivo principal da experimentação agrícola é fazer comparações dos efeitos de tratamentos aplicados nas amostras obtidas durante a implantação e acompanhamento do experimento. As principais fases da experimentação são: planejamento, execução, análise dos dados e interpretação dos resultados (NOGUEIRA, 1997).

O Planejamento de Experimentos (DOE - *Design Of Experiments*) é fundamental para se ter resultados estatísticos confiáveis e assim tomar decisões que promovam o sucesso do experimento. O ideal é que se consulte um estatístico para elaborar um plano experimental. Todo experimento deve começar com uma hipótese, ou seja, é preciso ter os objetivos do experimento bem definidos. Quando isso não ocorre, tem-se o risco de experimentos não conclusivos ou informações com resultados imprecisos (HOFFMANN; VIEIRA, 1989).

O Planejamento de Experimentos é uma técnica utilizada para definir quais dados, em que quantidade e em que condições devem ser coletados durante um determinado experimento. Buscando, basicamente, satisfazer dois grandes objetivos: a maior precisão estatística possível na resposta e o menor custo (HINKELMANN; KEMPTHORNE, 1994).

Basicamente, o planejamento envolve as seguintes etapas: formulação de hipóteses, definição dos fatores (variável independente) e seus respectivos níveis, definição da unidade experimental (parcela), definição do delineamento experimental, definição das variáveis resposta (variável dependente). Essas informações são fundamentais para a arquitetura RFlow, pois com elas será possível gerar os resultados estatísticos e a reprodução dos experimentos apoiados por *scripts*.

2.1.1 Formulação de hipóteses

Segundo Marconi e Lakatos (2010), a hipótese de uma pesquisa pode ser definida

como uma solução provisória, provável para um determinado problema, que tem um caráter explicativo daquela questão. A formulação da hipótese estatística tem como objetivo rejeitá-la ou validá-la. Por exemplo, um experimento de cultivares de feijão para verificar se um cultivar é melhor do que o outro com relação à incidência de pragas, a hipótese a ser formulada é:

"*Não existem diferenças significativas entre os seus efeitos."*

Portanto, quaisquer diferenças observadas são devidas a fatores não controlados, ou seja, ao acaso. Essa hipótese inicial recebe o nome de hipótese nula, e é representada por H_0, mas, se verificarmos a existência de diferenças discrepantes para os resultados dessa hipótese, podemos concluir que essas diferenças observadas são significativas, e rejeitamos a hipótese de nulidade em favor de uma outra hipótese, chamada de hipótese alternativa, e é representada por H_1 ou H_a (Quadro 1).

Quadro 1. Teste de Hipótese

Situação real	**Decisão**	
	H_0: Efeito não existe	H_a: Efeito existe
Efeito não existe	Correta	Incorreta (Erro tipo 1)
Efeito existe	Incorreta (Erro tipo 2)	Correta

O erro que corresponde à decisão incorreta de declarar que o efeito existe quando ele não existe é denominado erro tipo 1. O outro erro de decisão correspondente a declarar que o efeito não existe quando ele existe é denominado erro tipo 2.

2.1.2 Definição dos fatores e seus respectivos níveis

Fatores são as variáveis independentes do modelo matemático. Indicam o que está em comparação, por exemplo: fertilizantes, ração, métodos, variedade de cultivares, etc. Em um experimento, um fator pode ter várias divisões que são chamados de níveis. Refere-se a cada uma das alternativas de um fator em estudo para resolver um problema.

Exemplificando, tem-se o seguinte caso: um pesquisador deseja estudar o efeito de 2 variedades de feijão e 3 doses de nitrogênio. Neste caso trata-se de um experimento em fatorial 2x3, em que se tem dois fatores (variedade e dose de nitrogênio). O fator variedade tem 2 níveis e o fator nitrogênio tem 3 níveis. Os níveis do fator são denominados de tratamentos. No exemplo citado são aplicados 6 tratamentos (2 x 3). Os tratamentos ou fatores são classificados em quantitativos e qualitativos.

Quantitativos:

Doses: 0, 50, 150 kg/ha; dias após plantio: 30, 60, 90, etc...

Qualitativos:

Sexo (masculino e feminino); níveis de acidez (baixa, média, alta).

2.1.3 Definição da unidade experimental ou parcela

Unidade experimental ou parcela é a menor unidade de um experimento onde é realizada a aplicação do tratamento. É a unidade experimental que fornece os dados para serem avaliados. Em experimentos de campo, as unidades experimentais são denominadas parcelas. Como exemplo de unidade experimental ou parcela pode-se citar: um animal, um vaso contendo uma ou mais plantas, uma peça do motor, uma placa de Petri com meio de cultura, etc.

O número de parcelas é igual ao número de tratamentos vezes o número de repetições dos tratamentos.

I = número de Tratamentos J = número de Repetições

IJ = número de Unidades Experimentais ou número de Parcelas

Por exemplo, um experimento em fatorial com 3 x 4 x 2 e com 2 repetições. Neste exemplo, têm-se 3 fatores, 24 tratamentos e 48 parcelas.

2.1.4 Definição do delineamento experimental

É a maneira como os tratamentos foram distribuídos às unidades experimentais. A escolha correta do delineamento vai reduzir bastante o erro experimental. A análise de variância (ANAVA) é baseada no delineamento experimental utilizado. Um delineamento experimental deve ser planejado de maneira que a variação ao acaso seja reduzida o máximo possível. Os principais delineamentos experimentais são: delineamento inteiramente casualizado (DIC), delineamento em blocos casualizados (DBC), delineamento em quadrados latinos (DQL), delineamentos em blocos incompletos (por exemplo, os látices, blocos aumentados).

2.1.5 Escolha das variáveis resposta

Variáveis respostas ou variáveis dependentes ou apenas variáveis são parâmetros de saída resultantes de uma variação nas variáveis de entrada. Na escolha das variáveis resposta, o pesquisador deve ter certeza de que aquela variável realmente fornece informação útil sobre o experimento em estudo. Alguns exemplos de variáveis: peso de um animal, produção de grãos de milho; altura de plantas de eucalipto; teor de Ca, Mg e P em amostras de solo; etc. As variáveis resposta são classificadas da mesma forma que os fatores (tratamentos), em quantitativa e qualitativa.

2.1.6 Análise dos resultados

Após a obtenção dos dados, um Teste de Hipótese pode ser utilizado. Nessa fase são utilizados vários métodos da análise estatística. Testada a hipótese, o pesquisador interpreta o resultado, fazendo inferências sobre os resultados. Nesse momento, pode-se decidir pela elaboração de uma teoria (conclusões) ou por um novo experimento, completando o ciclo.

Maiores informações sobre esses princípios básicos podem ser localizados em livros textos especializados: (GOMES, 1996; GUERRA, M. J. & DONAIRE, D.,1991; BARROS NETO, J. C. *et. al*., 1995; BANZATTO, D. A.; KRONKA, S. N., 2006; entre outros).

2.2 E-*Science*

Ao longo da última década, muitas disciplinas evoluíram de registro de observações em simples cadernos de laboratório para a utilização de instrumentos capazes de gravar digitalmente muitos terabytes de dados em um único dia. Essa abundância de dados proporciona oportunidades sem precedentes para as novas descobertas científicas.

O termo e-Ciência ou *e-Science* significa o apoio ao pesquisador para o desenvolvimento de ciência em larga escala utilizando infraestrutura computacional (MATTOSO et al., 2008). Existem outras definições referentes à *e-Science* apresentadas pela comunidade científica. John Taylor foi o mentor do termo *e-Science* em 1999 quando era o diretor geral dos Conselhos de Pesquisa do Reino Unido. Este termo foi usado para o conjunto de ferramentas e tecnologias necessárias para suportar a pesquisa científica do século 21, caracterizada pela natureza colaborativa e multidisciplinar, pelo grande volume de dados e pelo importante papel da Tecnologia da Informação (Figura 1).

Segundo Vaz (2011), os grandes desafios impostos pelo processamento de crescentes volumes de dados e ainda distribuídos geograficamente exigem mais que somente automatizar a Ciência. São necessários profissionais com novas competências, novos métodos e uma infraestrutura robusta que permita realmente o trabalho colaborativo dos pesquisadores das mais variadas áreas e domínios. A Computação tem um papel fundamental nesse desafio, pois está presente em todos esses aspectos. A *e-Science* está inserida nesse contexto.

A *e-Science* é fundamental para a exploração e tratamento de dados demandados por telescópios, sensores e satélites distribuídos geograficamente. Além desses recursos tecnológicos, a área agrícola já utiliza VANTS, georreferenciamento, entre outros. O volume de dados gerados está na ordem de petabytes, e ainda, os usuários podem estar dispersos geograficamente ao redor do globo. Dessa forma, a *e-Science* deve construir a infraestrutura necessária para esses novos desafios (MATTOSO et al, 2008).

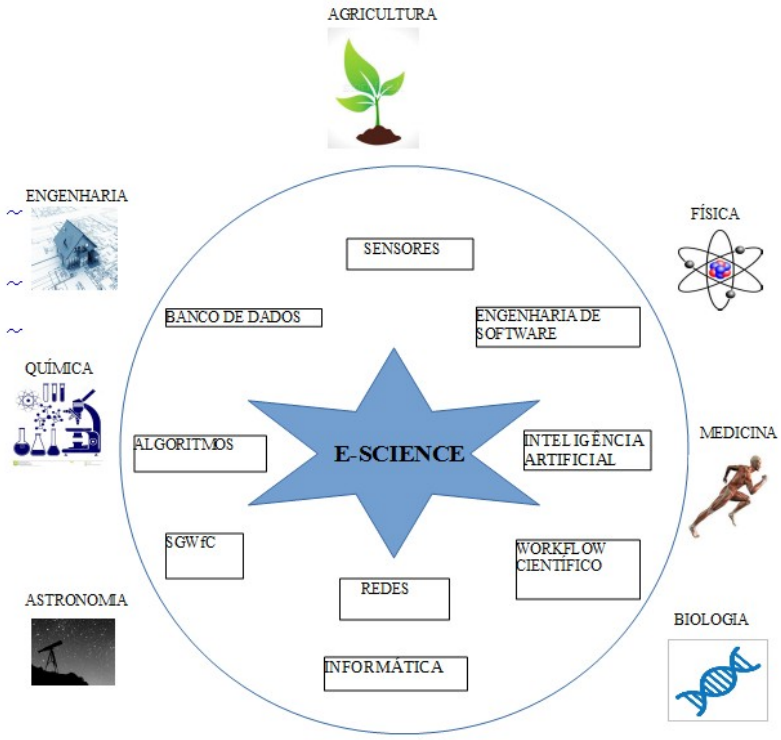

Figura 1. Representação das áreas comuns à *e-Science.*

No Brasil e no mundo já existem centros de pesquisa que se estruturam especificamente em torno de temas de *e-Science*. Por exemplo, em 2001, o Reino Unido lançou um programa pioneiro, que recebeu investimentos da ordem de 250 milhões de libras para estimular o desenvolvimento de *e-Science* em todos os campos de pesquisa (ATKINSON et al., 2009). Outro exemplo de sucesso é o *E-Science Institute* da Universidade de Washington (WASHINGTON, 2015), que foi criado em 2008 e tem trabalhado para criar a infraestrutura intelectual e computacional necessária para enfrentar os desafios da *e-Science.* O instituto desenvolve métodos e ferramentas computacionais avançadas para problemas do mundo real. Sua tarefa é procurar e envolver pesquisadores em todas as disciplinas em que as abordagens *e-Science* possam ter o maior impacto. Para garantir que os pesquisadores tenham acesso à infraestrutura física, o Instituto utiliza plataformas locais e remotas.

Um caso de sucesso brasileiro ocorre na UNICAMP, o *Center of Computation and Engineering Sciences* (CCES) (UNICAMP, 2015), que visa servir de centro multidisciplinar de classe mundial dedicada ao desenvolvimento e aplicação da ciência computacional avançada para os seguintes fins: formação de recursos humanos altamente qualificados e dedicados a enfrentar os desafios atuais em computação de alto desempenho e computação massiva em ciência e engenharia; promoção de parcerias entre o meio acadêmico e a indústria, educação científica e divulgação nessas áreas. O CCES tem como objetivos estudar os aspectos de alto desempenho e computação intensiva de dados aplicada às ciências físicas e engenharia química, mecânica e de materiais, biologia computacional e bioinformática, geofísica computacional e ciência da computação.

Construir uma plataforma de *e-Science* não é uma tarefa trivial e consome muitos recursos humanos, tecnológicos e financeiros. A própria Embrapa busca desenvolver estruturas computacionais para apoiar seus pesquisadores (VAZ, 2011), como é o caso da Embrapa Informática Agropecuária, uma unidade de pesquisa temática da Embrapa, cuja missão é "viabilizar soluções de pesquisa, desenvolvimento e inovação em tecnologia de informação para a sustentabilidade da agricultura em benefício da sociedade brasileira" (EMBRAPA INFORMÁTICA AGROPECUÁRIA, 2015a). Sua principal função é a de prover uma infraestrutura computacional para dar suporte às pesquisas desenvolvidas pela empresa e seus parceiros. A Embrapa Informática Agropecuária já desenvolve projetos de *e-Science*, um exemplo disso é o Laboratório Multiusuário de Bioinformática (EMBRAPA INFORMÁTICA AGROPECUÁRIA, 2015b), que disponibiliza recursos computacionais para os pesquisadores de toda Embrapa.

No entanto, na Embrapa, ainda não há uma plataforma corporativa de *e-Science* que atenda toda a cadeia de pesquisa (pesquisadores, parceiros, universidades, clientes, entre outros). Há muito ainda a ser feito. A proposta da arquitetura RFlow pode ser considerada uma pequena parcela neste grande esforço institucional.

2.3 Experimento Científico *in silico*

Durante muito tempo os pesquisadores reconheceram a teoria e experimentação em bancada como os paradigmas científicos básicos para entender os fenômenos da natureza. A experimentação científica sempre foi utilizada com o objetivo de adquirir novos conhecimentos ou corrigir e integrar conhecimentos previamente estabelecidos (WILSON, 1991).

Entretanto, um novo fenômeno vem ocorrendo com o avanço da computação nas últimas décadas, os experimentos científicos passaram a utilizar ferramentas computacionais para modelar ou simular fenômenos, minimizar falhas, reduzir custos experimentais, acelerar a sua execução. Esse fenômeno ocorre, talvez, em função do aumento da complexidade dos experimentos, já que seria quase que inviável em muitos casos a sua execução sem o apoio computacional, tais como, o mapeamento genético, a predição das mudanças de clima, a evolução das galáxias, a modelagem molecular, a simulação de sinais neurológicos, entre outros (Figura 2).

O aumento da complexidade dos experimentos científicos e do processamento crescente de volumes de dados têm demandado ferramentas computacionais e equipes multidisciplinares que possam atender a este novo cenário. De maneira que, além dos experimentos tradicionais como os experimentos *in vivo* (experimentos aplicados em campo) e *in vitro* (experimentos aplicados em laboratório), surgiram mais duas categorias de experimentos: *in virtuo* e *in silico*. Essas duas novas categorias de experimentos têm como características de serem baseadas em simulação e modelos computacionais. Nos experimentos *in silico*, os participantes e o ambiente são simulados. Nos experimentos *in virtuo*, os participantes reais interagem com o ambiente simulado (TRAVASSOS; BARROS, 2003).

De acordo com Anderson (2008), atualmente é possível fazer ciência analisando dados sem ter hipóteses prévias sobre eles. Por exemplo, com algoritmos computacionais, como sequenciamento genético, *data mining*, *data analytics,* entre outros, é possível encontrar padrões, mesmo que não haja teorias ou modelos sobre os dados analisados. Em muitos desses casos não é necessário seguir uma abordagem de hipótese, modelo e teste. Foi o que ocorreu para os vencedores do prêmio Nobel de Química de 2013, conferido a três químicos.

Todo o trabalho foi feito no computador, ou seja, não foi usada bancada de laboratório. Eles criaram métodos para que poderosos programas de computador sejam usados para entender e prever processos químicos. Isso permite a simulação das mais complexas reações químicas no computador. Por exemplo, simulações no computador de uma determinada droga para medir seu efeito antes de aplicar em um experimento *in vivo* (NOBELPRIZE, 2013).

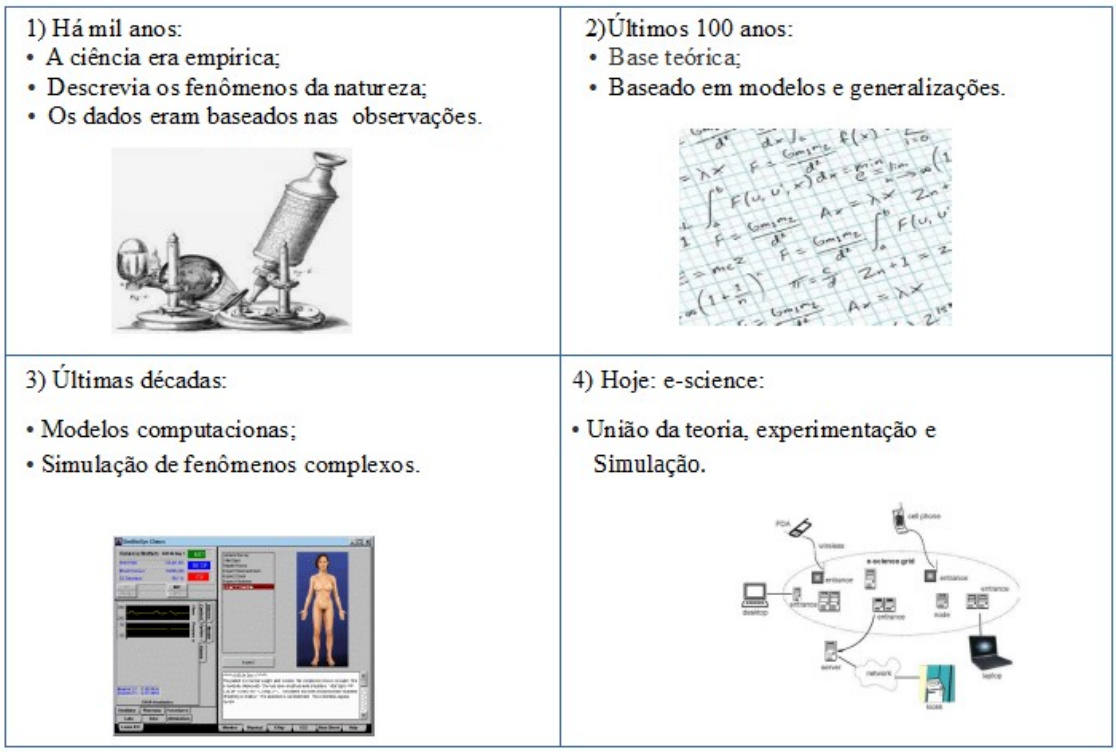

Figura 2. Evolução do método científico.

2.4 *Workflows* Científicos

O entrelaçamento entre o avanço do processamento de alto desempenho, da manipulação de crescentes volumes de dados científicos e a necessidade de gerência da proveniência em experimentos científicos *in silico* têm contribuído para o aumento sistemático do uso de *workflows* científicos em diversas áreas da Ciência (MATTOSO et al., 2009).

De acordo com Altintas et al. (2006), um *workflow* científico é uma sequência de passos ou tarefas onde trafegam dados e processos com o objetivo de solucionar um problema científico. Essa abordagem é baseada em experimentos científicos apoiada pela Ciência da Computação, denominada *e-Science*. Representa um estudo baseado em simulação e segue um conjunto de fases: composição, execução, análise e proveniência (OINN et al., 2007).

Os padrões de *workflows* (AALST et al., 2003; RUSSELL et al., 2006) definem as características de modelagem do *workflow*. Através deles é possível verificar que a saída de uma atividade torna-se a entrada para a atividade seguinte. Isso é feito de forma sistemática

até que a resposta ou o objetivo seja atingido. Na Figura 3 é mostrado o fragmento de um *Workflow* concreto implementado no SGWfC Kepler. Cada componente representa uma atividade (ator) relacionada à leitura e gravação em um banco de dados.

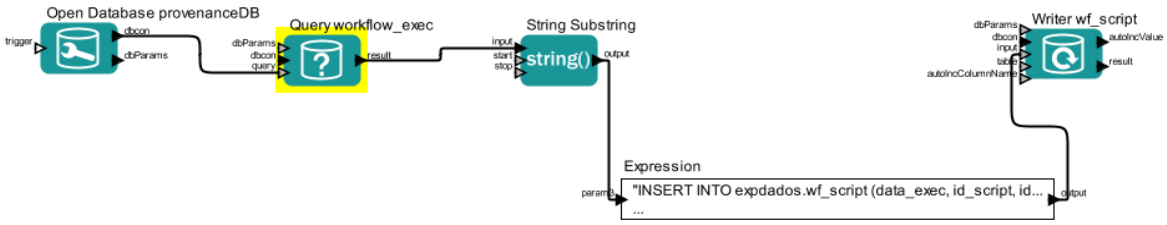

Figura 3. Exemplo de *workflow* científico concreto feito no SGWfC Kepler. Realiza atividades de acesso e gravação em banco de dados.

2.4.1 *Workflows* Abstratos

Os *workflows* abstratos estão em um nível mais alto, ou seja, são as representações conceituais de cada passo de um experimento. São elaborados na fase de composição, que é responsável pela concepção e encadeamento dos processos que constituem o experimento (DEELMAN et al., 2009). Traçando um paralelo com a experimentação agrícola, seria a fase de planejamento do experimento, que antecede a materialização do *workflow* em programas e dados executados no computador por um SGWfC. O planejamento do experimento agrícola está associado à proveniência prospectiva.

2.4.2 *Workflows* Concretos

É a materialização do *workflow* abstrato, ou seja, são incorporados os recursos computacionais (programas e dados) para atender um determinado experimento científico e assim ser executado em um SGWfC. O *workflow* concreto é uma instância específica de um *workflow* abstrato para resolver um determinado problema (MATTOSO et al., 2009).

No contexto da experimentação agrícola, o *workflow* concreto será o responsável pela execução e pelo histórico de execuções das instâncias do experimento científico no SGWfC, mais especificamente as análises e resultados estatísticos. Está associado a proveniência retrospectiva.

2.5 *Workflows* Estatísticos (baseados nos *scripts* R)

A utilização de *workflows* científicos com foco no processamento estatístico também vem crescendo ao longo dos últimos anos. Estes se caracterizam pela manipulação de grandes volumes de dados e por executarem sofisticadas análises estatísticas através da incorporação de recursos (funções, algoritmos e métodos) disponíveis em sistemas estatísticos (SPSS, SAS, Statistica, Mapple, MathLab, Weka, R, entre outros (MAIR; DE LEEUW, 2010)).

Segundo Rumar e Wainer (2005) e Ranabahu et al. (2011), os *workflows* estatísticos

podem processar bases de dados locais ou remotas e são codificados através de linguagens próprias dos sistemas estatísticos sob a forma de complexos *scripts*.

Os *workflows* estatísticos são geralmente desenvolvidos por pesquisadores que possuem grande experiência em estatística e limitada capacidade de programação de computadores e conhecimentos de distribuição de processamento em múltiplos nós. Por esse motivo, requerem especialistas e possuem um alto custo de desenvolvimento. São em geral de domínios específicos e difíceis de serem reutilizados e compartilhados por terceiros (KIRCHKAMP, 2014). Atualmente, não estão disponíveis na literatura muitos trabalhos que relacionam proveniência, SGWfC e sistemas estatísticos através de meta-*workflows* reutilizáveis que encapsulam *scripts* sem a necessidade de alteração em seus códigos-fonte (NASCIMENTO; CRUZ, 2013).

2.6 Sistema Gerenciador de *Workflow* Científico (SGWfC)

Com o propósito de automatizar e gerenciar a construção e execução dos *workflows* científicos foram desenvolvidas ferramentas computacionais denominadas sistemas de gerência de *workflows* científicos (SGWfC) (HEY et al., 2009).

O crescimento cada vez maior do volume de dados na pesquisa aliada a diversas formas de explorá-los e mantê-los, como dados distribuídos, sensores, Web e satélites têm demandado ferramentas que suportem todo o ciclo de pesquisa, desde a captura e curadoria dos dados até sua análise e visualização. Uma dessas classes de ferramentas é o SGWfC (GRAY, 2009).

O SGWfC é o software responsável pela execução dos *workflows* científicos do tipo concreto. A automação de *workflows* pode fornecer as informações necessárias para a reprodutibilidade científica, a derivação, a geração de dados e o compartilhamento de resultados em um ambiente de pesquisa colaborativo (OINN et al*.,* 2007).

Segundo Silva (2011), o SGWfC possui toda infraestrutura necessária para executar, definir e monitorar as execuções dos *workflows* científicos tanto local quanto remotamente. Os SGWfC's oferecem aos pesquisadores interfaces gráficas que facilitam o desenvolvimento dos *workflows* bem como controle de execução, captura de proveniência e seu monitoramento. Possibilitam que os pesquisadores realizem uma espécie de programação em alto nível, através do encadeamento de processos científicos (ou atividades) que seguem uma sequência lógica do experimento.

Atualmente, existem dezenas de SGWfC disponíveis (TALIA, 2013; LITTAUER et al., 2012), alguns de domínio específico, como o Galaxy (GIARDINE et al., 2005) (https://usegalaxy.org/), para a área biomédica o Taverna (OINN, 2006) (http://www.taverna.org.uk/) e o Tavaxy (ABOUELHODA et al., 2012) (http://tavaxy.org/) para a bioinformática, o Weka4WS para tarefas de mineração de dados (TALIA; TRUNFIO; VERTA, 2005), o DIS3GNO (CESARIO et al., 2011) e o Triana (TAYLOR, 2004) para ambientes de grade de computadores e o sistema Pegasus para ambientes multiplataforma de *High Performance Computing* (HPC) e nuvens de computadores (DEELMAN et al., 2005; NAGAVARAM et al., 2011; VÖCKLER et al., 2011). Além desses, existem SGWfC semânticos (ZHAO; PASCHKE, 2012) que incorporam características comuns à Web Semântica, tais como processamento de dados *Resource Description Framework* (RDF), uso de ontologias em *Web Ontology Language* (OWL), entre outros. Além desses sistemas,

existem os de propósito geral que podem ser facilmente instalados em simples *desktops*, dentre eles destacam-se o VisTrails (CALLAHAN et al., 2006) e o Kepler (LUDÄSCHER et al., 2006).

Nas próximas subseções são descritos três SGWfC de propósito geral que possuem grande aceitação na comunidade científica: Kepler, VisTrails e Taverna. Todos são sistemas centralizados, que suportam *workflows* científicos do tipo grafo direcionado acíclico (do inglês, DAG) e elaborados sob o paradigma do software livre. Isso permite que a comunidade de desenvolvedores estejam sempre evoluindo a ferramenta. Posteriormente, será apresentada uma matriz de comparação que justifica a escolha do Kepler como SGWfC integrante da arquitetura RFlow.

2.6.1 SGWfC Kepler

O Kepler é mantido pela colaboração de vários projetos, que é liderada por uma equipe composta pelas instituições: UC Davis, UC Santa Barbara, e UC San Diego. É um aplicativo baseado em Java que executa nos sistemas operacionais Windows, Mac OS X e Linux. Está na versão 2.4. Pode ser baixado de https://kepler-project.org/.

O Kepler é um software aplicado para análise e modelagem de dados científicos. Ele simplifica o esforço do pesquisador para criar, alterar e executar *workflow*s concretos. É um sistema consolidado no meio científico, de código aberto, feito em Java e construído sobre o motor de execução PtolemyII (ambiente para experimentos com simulações heterogêneas). Visa atender a diversos domínios do conhecimento. Ele tem como meta desenvolver soluções genéricas tanto para o processamento de *workflows* científicos quanto para os desafios de integração de aplicações científicas (LUDÄSCHER et al., 2006).

O Kepler é um ambiente de desenvolvimento específico para análise, manipulação, modelagem e simulação de *workflows* científicos. Fornece aos usuários uma ferramenta de fácil uso e que simplifica a criação e a execução de *workflows* científicos, aumentando a produtividade dos pesquisadores (MATTOSO; CRUZ, 2008).

O desenvolvimento de *workflows* no Kepler é com base nos princípios da orientação a atores. Utiliza o conceito de diretor/ator para representar os componentes do *workflow* e a comunicação entre eles. Assim como em um filme, o diretor do Kepler comanda o seu elenco (atores), especificando quando cada um deve agir e como estão conectados entre si. Já o papel dos atores é processar os dados disponíveis nas suas portas de entrada e disponibilizar os resultados na porta de saída.

O Kepler é rico em componentes, com aplicação em diversas áreas, por exemplo, em bioinformática, estatística, biologia, ecologia, etc. Os dados e *workflows* podem ser compartilhados e usados por pesquisadores localizados em qualquer lugar do planeta que tenha acesso à Web (LUDÄSCHER et al., 2006).

O pacote que permite a captura de proveniência é o *Provenance Recorder* (*PR*) (ALTINTAS et al, 2006). O PR será visto com mais detalhes no Capítulo de Materiais e Métodos.

2.6.2 SGWfC Taverna

É um sistema com forte aplicação na bioinformática, tendo inúmeras ferramentas gráficas. Foi concebido pela equipe myGrid (MYGRID, 2008) e é financiado por FP7 referente aos projetos BioVeL, SCAPE e Wf4Ever. Atualmente está na versão 2.5 (http://www.taverna.org.uk/) e foi desenvolvido em Java. Executa nos seguintes sistemas operacionais: Windows 64-bit, Windows 32-bit, Mac OS X 64-bit, Linux 64-bit Debian (Ubuntu 12.04.4, Bio-Linux 7), Linux 64-bit Redhat (Fedora 20). É distribuído em quatro plataformas: Taverna Workbench (aplicativo cliente de *desktop*), a ferramenta de linha de comando (execução do *workflow* em um terminal), o *Server* (para execução remota do *workflow*) e o *plugin* Player (interface da *Web* para a apresentação de *workflows* para a execução remota) (HULL et al., 2006).

O Taverna pode capturar proveniência das execuções de *workflows*, incluindo iterações de processadores individuais e suas entradas e saídas. Essa proveniência é mantida em um banco de dados interno. Utiliza o *plugin* Taverna-PROV para proveniência na execução de *workflows*, incluindo a saída e valores intermediários. No rastreamento de proveniência é utilizado PROV-O RDF gráfico, que pode ser consultado usando SPARQL e processadas com outras ferramentas de Prov, tais como o PV Toolbox (HULL et al., 2006).

O Taverna permite fazer a ciência colaborativa, ou seja, o compartilhamento de seus *workflows* através de busca e *download* no *sítio* myExperiment (MYGRID, 2008).

2.6.3 SGWfC VisTrails

O VisTrails foi desenvolvido em Python/Qt e lançado em outubro de 2007. Está na versão v2.2 e executa nos sistemas operacionais Mac, Linux e Windows. Nos novos lançamentos vem com vários pacotes, incluindo VTK, matplotlib e ImageMagick (VISTRAILS, 2015).

O foco do VisTrails são experimentos de simulação, exploração de dados e visualização. Os *workflows* podem ser construídos de diversas formas e ao longo do desenvolvimento podem ser comparados e aperfeiçoados. Isso é possível pois VisTrails gera um histórico detalhado de todos os passos seguidos na construção do *workflow*. Essas informações são mantidas como arquivos XML ou em um banco de dados relacional, e permite que os usuários naveguem entre versões de *workflows* e que façam alterações no *workflow* sem perder qualquer informação. Permite, também, comparar diferentes *workflows* e as ações que conduziram a uma nova versão (FREIRE et al., 2008).

Um ponto importante no VisTrails é a questão de usabilidade e visualização, já que a maioria dos SGWfC têm interfaces difíceis de manusear, principalmente por usuários que não têm experiência em programação. Essas facilidades de interação permitem que o usuário visualize todas as diferentes versões obtidas, explorando variações no experimento com mínimo de esforço (ELLKVIST et al., 2008).

2.6.4 Matriz de Decisão

O SGWfC é o componente fundamental na arquitetura RFlow. Para haver uma boa integração e o mínimo possível de intervenção nos componentes da arquitetura, foram testadas e observadas algumas características necessárias para tal. Pelos testes e observações

nos três SGWfC (Kepler, Taverna e VisTrails), verificou-se que todos apresentam as características necessárias para a integração na RFlow. Entretanto, nesta primeira fase do desenvolvimento da arquitetura prima-se pela ferramenta que terá menos intervenções ou adaptações.

A métrica foi definida da seguinte forma: 1 – sem adaptações ou intervenção mínima; 2 – com adaptações. O Quadro 2 mostra o resultado:

Quadro 2. Matriz de decisão

Critério	**Taverna**	**VisTrails**	**Kepler**
Integração com o R	1	2	1
Proveniência retrospectiva em banco de dados	2	2	1
Execução via linha de comando	1	2	1
Suporte a execução remota	1	1	1
Utilização do PostgreSQL	2	2	1
Total	**7**	**9**	**5**

Pelos resultados apresentados, verifica-se que o Kepler é o componente mais apropriado nesta fase inicial da implementação da arquitetura RFlow. A sua integração foi bem natural, pois conforme visto na Matriz, não houve nenhuma intervenção ou esta foi mínima para o bom funcionamento da plataforma. Será visto com mais detalhes no próximo capítulo.

2.7 Linguagem R

O R é uma linguagem de código aberto, sob a licença GNU. Possui uma plataforma de desenvolvimento que interage com os usuários através de linha de comandos, semelhante ao DOS e UNIX. Foi iniciada a partir de 1993, derivada da linguagem S, desenvolvida inicialmente por John Chambers, Rick Becker e Allan Wilks dos Laboratórios Bell (R DEVELOPMENT CORE TEAM, 2012), da qual existe uma versão comercial chamada S-PLUS.

O R é o software estatístico responsável pela interpretação e execução dos *scripts* R. Possui um ambiente interativo de programação maduro e largamente utilizado, que permite a codificação de *scripts* capazes de executar sofisticados processamentos estatísticos (CRAWLEY, 2002) e (CHAMBERS, 2008).

O R é amplamente difundido nas comunidades científicas brasileira e internacional, em especial nos domínios da agronomia, meteorologia, ecologia, geoprocessamento, bioinformática, biologia molecular, entre outros. Com o R é possível escrever desde pequenas linhas de comandos até sofisticados *scripts* desenvolvidos de modo *ad hoc*. Estes *scripts*

podem ser executados pelo sistema R em momentos diferentes. Entretanto, o R apresenta problemas de não gerar nem capturar proveniência acerca dos processamentos e análises realizadas pelos pesquisadores.

Constantemente estão sendo adicionados novos pacotes ao R, o que torna a linguagem cada vez mais poderosa e com mais recursos para solucionar problemas dos mais variados.

2.7.1 Tipos de *Script* R

Graças à flexibilidade da linguagem R é possível criar diversos tipos de *scripts* R, a saber:

Tipo 1 - *Scripts* com entrada de dados via console. São *scripts* que permitem que o usuário interaja com o sistema R e faça testes com dados e funções básicas.

Tipo 2 - *Scripts* com função de leitura de arquivos CSV. Importar dados de arquivos do tipo CSV e armazená-los localmente.

Exemplo: *mydata <- read.csv (" filename.txt ")*

Tipo 3 - *Scripts* com carga de arquivos de dados locais. Importar dados de arquivos de pacotes estatísticos e de SGBD relacionais ou NoSQL.

Tipo 4 - *Scripts* com carga de arquivos de dados remotos. Importar dados de arquivos em sítios remotos.

Exemplo: mydata <- read.csv("http://bit.ly/10ER84j")

O Sistema R e seus diversos tipos de *scripts* podem ser associados com diversos SGWfC. Por exemplo, a associação entre o R e o Taverna dá-se pelo componente RShell. No Kepler, ela ocorre mediante a construção de *workflows* que incorporam atores e diretores R-específicos (por exemplo, ReadTable, RandomNormal, ANOVA, Correlation, LinearModel, RMean, Rmedian, Rquantile, Summary, Barplot, Boxplot, RExpression, Scatterplot) (LUDÄSCHER et al*,* 2006).

No SGWfC VisTrails não estão disponíveis módulos R-específicos, logo a associação se dá através de codificação direta de módulos em *workflows* que invocam os recursos estatísticos do sistema R.

2.8 Proveniência

O problema da proveniência de dados foi inicialmente caracterizado por Buneman et al. (2001). Para o autor, a proveniência de dados, também chamada de linhagem, genealogia ou *pedigree*, consiste na descrição das origens de um item de dado e do processo pelo qual foi produzido. A proveniência dos dados auxilia a formar uma visão da qualidade, da validade e de quão recente é a informação. No escopo de *workflows* científicos, a proveniência de dados fornece informação histórica acerca dos dados manipulados a partir de suas fontes originais (SIMMHAN et al*.,* 2005). Essa informação pregressa descreve os dados que foram gerados, apresentando os seus processos de transformação a partir de dados primários e intermediários. Nesse cenário, as informações de proveniência agregam valor de forma significativa no processo de gerência dos resultados obtidos computacionalmente pelos pesquisadores.

A gestão da proveniência tem por objetivo servir de auxílio na busca de respostas a inúmeras indagações concernentes a um experimento científico, dentre as quais é possível citar: Que análises estão disponíveis? Como unificar e resumir os conhecimentos gerados? Como consultar uma base de experimentos? E muitas outras questões importantes nesse contexto (MATTOSO et al*.,* 2008) (CRUZ et al, 2009).

De acordo com Cohen et al. (2006), Freire et al. (2008) e Cruz (2011), a proveniência tem granularidades distintas e pode ser de diversos tipos, e classificada inicialmente como prospectiva ou retrospectiva. O primeiro tipo captura o processo de especificação de tarefas computacionais do *workflow* (programa, atividade, etc.), enquanto o segundo tipo captura as tarefas executadas, os dados e parâmetros utilizados, além das informações sobre o ambiente utilizado para derivar um resultado científico, consistindo em um tipo de histórico estruturado e detalhado sobre a execução de tarefas computacionais.

Alguns sistemas de coleta de proveniência usam estruturas internas do SGWfC para coletar a proveniência (CALLAHAN et al., 2006; ZHAO et al., 2004), normalmente em ambientes centralizados, e outros (ALTINTAS, BARNEY, JAEGER-FRANK, 2006; BITON, BOULAKIA, DAVIDSON, 2007; MARINHO, MURTA, WERNER, 2010) utilizam serviços externos, que são mais genéricos e voltados para ambientes distribuídos e heterogêneos. No caso do Kepler, é utilizado uma estrutura interna: o ator *Provenance Recorder*, que será discutido com mais detalhes no decorrer da dissertação.

Neste trabalho, a proveniência de dados tem um papel primordial, pois é a partir de seus mecanismos de coleta que será possível gerar um histórico de execuções dos *workflows,* rastrear os dados, reproduzir os experimentos, monitorar as alterações feitas nos metadados, visualização da evolução do *workflow*, entre outras funções.

Muitas vezes é necessário trocar informações de proveniência ou mesmo exportar os dados para outro SGWfC, nesse caso é necessário que haja um padrão de modelo de proveniência. Os metamodelos de proveniência mais usados são: *Open Provenance Model* (OPM) (MOREAU et al., 2008) e PROV (MOREAU et al., 2011), este último homologado pelo W3C (*World Wide Web Consortium*) (W3C, 2012). Os dois modelos de proveniência capturam somente a proveniência retrospectiva, ou seja, o histórico de execuções do *workflow*, por exemplo, Quem executou? Qual a máquina? qual *workflow*? Quais dados foram utilizados? Quais arquivos foram produzidos? Tempo inicial e final? etc.

O metamodelo OPM, segundo Moreau et al. (2007) possui as seguintes premissas: (i) possibilitar a representação digital de qualquer tipo de aplicação de proveniência; (ii) ser independente de tecnologia; (iii) permitir a troca de informações de proveniência entre sistemas.

O OPM representa os dados de proveniência através de um grafo causal, que registra o histórico do fluxo de execução do *workflow*. É baseado em três entidades básica: Artefato (representação imutável de um dado, objeto físico ou digital), Processo (ação ou conjunto de ações realizadas em artefatos ou causadas por eles que resultam em novos artefatos) e Agente (entidade contextual que age sobre um processo: habilitando, facilitando, controlando e afetando sua execução).

O metamodelo PROV é o mais atual e tem como alvo principal as aplicações para o ambiente Web. Os três elementos principais são: Entidade, Agente e Atividade. Apesar de ser mais recente que o OPM, já está bastante difundido na literatura. Utiliza os mesmos princípios do OPM, porém com mais detalhamentos e novos relacionamentos. O grupo de trabalho

produziu diversas especificações para o modelo PROV-DM. Cada especificação tem uma peculiaridade, conforme descrito a seguir:

PROV-DM: tem como principal função descrever as pessoas, entidades e atividades envolvidas na produção de um artefato de dado ou de um objeto qualquer (visão geral).

PROV-CONSTRAINTS: define um conjunto de restrições aplicadas ao modelo PROV-DM.

PROV-N: define uma notação para proveniência destinada ao uso em linguagem descritiva.

PROV-O: define uma ontologia OWL-RL que permite mapear o modelo PROV-DM para o padrão RDF.

PROV-AQ: define mecanismos de acesso e consulta de proveniência.

PROV-PRIMER: apresenta uma introdução ao modelo de proveniência.

PROV-SEM: define uma semântica formal do modelo PROV-DM.

PROV-XML: esquema XML para o PROV-DM.

Os SGWfC VisTrails e o Taverna já utilizam o metamodelo PROV. Porém, o Kepler 2.4 continua usando o metamodelo OPM. Em função disso, optou-se em mapear as tabelas do SisGExp para o metamodelo OPM, mantendo a compatibilidade com o sistema de coleta de proveniência do SGWfC Kepler. Essa abordagem não limita o trabalho, visto que os metamodelos guardam grandes semelhanças uns com os outros. O SisGExp é o componente responsável pela coleta da proveniência prospectiva, que representa os passos a serem seguidos até a execução do *workflow* estatístico. Para manter a compatibilidade com o metamodelo OPM, o mapeamento do modelo de dados do SisGExp seguiu a seguinte premissa para as entidades principais: Experimento compatível com Artefato, Atividade compatível com Processo e Usuário compatível com Agente.

2.9 Coleta de Proveniência no Kepler: *Provenance Recorder* (PR)

O coletor de proveniência utilizado nesta dissertação é o *Provenance Recorder* (PR) (ALTINTAS et al., 2006). Sua escolha se deve aos seguintes motivos:

(i) mecanismo de simples utilização (representado pelo ator *Provenance Recorder* adicionado ao meta-*workflow*);

(ii) capacidade de registrar a proveniência dos dados de entrada, saída e intermediários, além das definições do *workflow* (diretores, atores, sub-*workflows*, portas, tokens, *script* R utilizado) e informações de *timestamp* relacionadas a sua execução;

(iii) capacidade para armazenar proveniência de baixa granulosidade (detalhe do dado) e oferecer facilidades de conectividade tanto com arquivos XML ou Oracle, MySQL, PostgreSQL, HSQL;

(iv) por fim, o sistema é compatível com a especificação OPM.

O ator PR trabalha de maneira integrada com o Kepler (captura proveniência no nível de *workflow*). A coleta de informações de proveniência retrospectiva é realizada a cada instância de *workflow* concreto executada. Através das portas e canais das atividades do *workflow* em execução são emitidos sinais de leitura ou gravação, o que permite o PR registrar os parâmetros, dados e metadados em banco de dados ou em arquivo no formato XML. Dessa maneira, a

captura da proveniência retrospectiva é feita de forma consistente (BOWERS et al., 2006).

2.10 Trabalhos Relacionados

Atualmente existem poucos trabalhos que correlacionam a coleta transparente de proveniência, *workflows* científicos e sistemas estatísticos através do uso de meta-*workflows* reutilizáveis que preservam os *scripts* legados sem a necessidade de alteração em seus códigos-fonte. Ao contrário, as alternativas atuais apontam soluções na direção oposta (HIGGINS, 2007). Por exemplo, o SGWfC Kepler oferece um conjunto de atores R-específicos que precisam ser explicitamente modelados sob a forma de atividades concretas no *workflow* para que invoquem os recursos do sistema R. Essa abordagem não é tecnologicamente neutra, pois exigem razoáveis esforços de programação por parte dos pesquisadores e abandono ou substituição dos *scripts R* legados.

Uma alternativa que vem ganhando corpo nos últimos anos é a incorporação de recursos de proveniência aos sistemas estatísticos. Silles e Runnalls (2010) e Runnalls (2013) propõem a refatoração do código do sistema R para que incorpore recursos de proveniência no seu motor de execução. Eles apresentaram uma variante do R denominada CXXR. O sistema já oferece algum tipo de coleta de proveniência retrospectiva sob a forma de *logs* de execução. No entanto, ainda está em desenvolvimento, e não possui todos os recursos de um SGWfC.

A maioria das universidades públicas e institutos de pesquisa brasileiros <http://www.periodicos.capes.gov.br/> <http://www.alice.cnptia.embrapa.br/> possuem armazenados em seus repositórios institucionais sua produção científica (artigos, dissertações, teses, notas técnicas, etc), onde é possível consultar e em alguns casos baixar os documentos para o computador do pesquisador. Entretanto, ainda não há a possibilidade do pesquisador validar os dados contidos nestes documentos ou mesmo (re)executar os experimentos apoiados por *workflows* registrados nessas pesquisas. Geralmente o que ocorre é a (re)digitação ou recarga dos dados descritos no documento para um novo ambiente de experimentação, o que nem sempre é possível ou viável de ser realizado. A arquitetura RFlow pode reduzir essa distância, uma vez que é capaz de armazenar os dados, os *scripts* e as publicações correlacionadas.

Na literatura já existem sistemas que oferecem repositórios de *workflows* que permitem que os pesquisadores acessem remotamente e incluam novos *workflows*, como por exemplo o myExperiment (GOBLE et al*.,* 2010) e o CrowdLabs (MATES et al*.,* 2011), porém essas alternativas não são capazes de encapsular de modo transparente os *scripts* R legados através de um meta-*workflow* genérico, nem oferecem suporte aos *workflows* desenvolvidos no SGWfC Kepler.

Novas abordagens para capturar proveniência na execução de *scripts* R estão surgindo. Se caracterizam por não utilizarem SGWfC para serem executados. Por exemplo:

a) RDataTracker (LERNER; BOOSE, 2014): possui ferramentas que permitem aos pesquisadores coletar, visualizar e consultar proveniência diretamente da linguagem estatística R. É composta por uma biblioteca de funções R que pode ser baixada e instalada como um pacote do R, e um programa Java para visualizar os grafos gerados pela ferramenta. É não intrusivo no *script* R.

b) RmarkDown (BAUMER et al., 2014): é uma ferramenta que funciona dentro do

ambiente rstudio (http://www.rstudio.com/). Pode ser instalado através do próprio ambiente do rstudio. O usuário precisa intervir no *script* R para que haja a coleta de proveniência. Na execução do código RMarkdown é produzido um arquivo *html* com os comentários do usuário, o código que foi executado e a saída da execução.

c) yesWorkflow (MCPHILLIPS et al., 2015): é formada por um conjunto de ferramentas integradas, que tem como base o Java. Permite aos pesquisadores anotar nos *scripts* existentes com comentários especiais que revelam os módulos computacionais e os fluxos de dados implícitos nos *scripts*. Durante a execução do *script* é identificada a marcação especial no *script* e com isso são produzidas representações gráficas do *script*.

O Quadro 3 apresenta as principais características de cada trabalho e compara com a arquitetura RFlow. O trabalho marcado com "X" representa a existência da característica.

Quadro 3. Comparação entre trabalhos relacionados

Característica	**RFlow**	**CXXR**	**RDataTracker**	**RmarkDown**	**yesWorkflow**	**Kepler**	**myExperiment**
Proveniência prospectiva	x	x	x	x	x		
Proveniência retrospectiva	x	x	x	x	x	x	x
Workflow científico	x					x	x
Vinculação da publicação com experimento científico	x						
Repositório de *scripts* R	x	x		x		x	x
Sem intrusão no *script* R	x	x	x				
Compartilhamento de *scripts* na Web	x			x	x		x

O Quadro 3 mostra que a arquitetura RFlow tem uma área de atuação maior que os outros trabalhos, pois, além da arquitetura estar alinhada com os conceitos já presentes na literatura e operacionalizar o repositório de descritores, experimentos e dados, em moldes assemelhados aos dos repositórios citados anteriormente, ela tem como diferencial permitir a validação dos dados dos experimentos através da reprodução dos resultados estatísticos (baseados em *scripts* R) *online* e ainda vincular as publicações com os experimentos anteriormente coletados e cadastrados pelo aplicativo SisGExp, componente da RFlow.

Outro aspecto relevante é que a RFlow gera proveniência retrospectiva das execuções dos *scripts* R sem haver necessidade do pesquisador fazer modificações no *script.* Isso aumenta a produtividade das pesquisas, pois o pesquisador foca em sua atividade-fim e não precisa se preocupar em adequar o *script* R para conseguir executar na arquitetura RFlow.

A arquitetura RFlow está baseada na Web, isso possibilita que o pesquisador possa compartilhar seus *scripts*, dados brutos, resultados estatísticos e publicações científicas entre

seus parceiros e outros pesquisadores do globo. É importante frisar que o compartilhamento dos dados de pesquisa deve ser autorizado pelo autor dos dados.

Portanto, verifica-se que esse trabalho está alinhado com a temática da *e-Science*, uma vez que utiliza ferramentas da tecnologia da informação, por exemplo, *workflow* científico, SGWfC, banco de dados, entre outras, para resolver desafios em vários domínios da ciência, como a agropecuária, estatística, ciência da computação, entre outros.

3 MATERIAIS E MÉTODOS

Segundo Ferrari (1982), a metodologia de pesquisa apresenta o caminho que determina o resultado a ser encontrado. A ordenação dos procedimentos de maneira lógica possibilita a reprodutibilidade do experimento. Permite compreender não apenas os resultados, mas o processo da própria investigação científica.

Neste capítulo são apresentados os materiais e métodos utilizados na concepção da arquitetura RFlow. A arquitetura deve prover camadas de interoperabilidade capaz de interagir com o SGWfC Kepler, Sistema R, banco de dados PostgreSQL e o aplicativo Web SisGExp, tendo como finalidade capturar as informações de proveniência geradas a partir da execução dos *scripts* R legados e anotações realizadas pelos usuários no SisGExp.

Também são descritos alguns experimentos executados pelo SisGExp com o objetivo de avaliar a capacidade da arquitetura de coletar proveniência prospectiva e retrospectiva bem como identificar necessidades de ajustes e aprimoramentos na mesma.

Este capítulo está organizado da seguinte forma: a seção 3.1 descreve os materiais; a seção 3.2 descreve os métodos utilizados para especificar a arquitetura; a seção 3.3 informa quais são as categorias de experimentos avaliadas na arquitetura RFlow.

3.1 Materiais

O trabalho está baseado nos conceitos de *e-Science*, uma vez que utiliza *workflow* científico, SGWfC, banco de dados, métodos de proveniência, entre outros, para especificar a arquitetura RFlow.

A arquitetura é multiplataforma, ou seja, pode ser implementada nos sistemas operacionais Linux e/ou Windows.

Os softwares utilizados para implementar a RFlow são: SGBD PostgreSQL 9.3 (POSTGRESQL, 2009), SGWfC Kepler 2.4 (ALTINTAS et al., 2004), Sistema R 3.1.1 (R DEVELOPMENT CORE TEAM, 2012), Java JDK 1.7 (ORACLE, 2014), Servidor de Aplicação GlassFish 4.0 (ORACLE, 2014). Para desenvolver o SisGExp foi utilizada a IDE (ambiente de desenvolvimento) Netbeans 8.0 (ORACLE, 2014).

Além desses softwares, foram utilizadas algumas bibliotecas de domínio público para otimizar o desenvolvimento do aplicativo SisGExp. A saber: Primefaces 5.2 (PRIMEFACES, 2009), commons-fileupload-1.3.1 (APACHE, 2014) e commons-io-2.4 (APACHE, 2014).

3.2 Métodos

Nas subseções abaixo são descritas as configurações e especificações realizadas nos softwares (materiais) para implementar a arquitetura RFlow. Foram realizados vários testes e ajustes de configurações para conseguir o melhor desempenho no processamento dos experimentos.

3.2.1 Banco de Dados provenanceDB

O Banco de Dados (BD) provenanceDB é capaz de armazenar os dois tipos de proveniência (prospectiva e retrospectiva), ele foi implementado com o SGBD PostgreSQL versão 9.3. No BD existem dois esquemas:

α) Esquema *public* (padrão), onde estão armazenadas as tabelas do SGWfC Kepler. O esquema será responsável pela coleta da proveniência retrospectiva;

β) Esquema *expdados* (projetado e desenvolvido), possui a estrutura das tabelas do aplicativo SisGExp. O esquema será responsável pela coleta da proveniência prospectiva.

O modelo de dados utilizado no esquema *expdados* é uma variação baseada no metamodelo OPM, uma vez que o Kepler utiliza esse metamodelo como padrão e assim preferiu-se manter a compatibilidade da RFlow com esse padrão. Os dois esquemas do provenanceDB serão apresentados e discutidos no Capítulo 4.

3.2.2 Configuração do *Provenance Recorder* (PR)

A configuração padrão do PR referente ao registro de proveniência retrospectiva de dados na execução dos *workflows* do Kepler é o banco de dados HSQL. Se o ícone <**P**> na barra de comandos do Kepler está na cor verde, indica que a proveniência de dados habilitada no Kepler é a genérica, e vermelha para não habilitada (Figura 4). Para mudar essa configuração deve-se alterar o arquivo "configuration.xml" que fica na pasta "\KeplerData\kepler.modules\provenance-2.4.1\resoures\configurations". Essa alteração afetará todas as execuções de *workflows* (Figura 5).

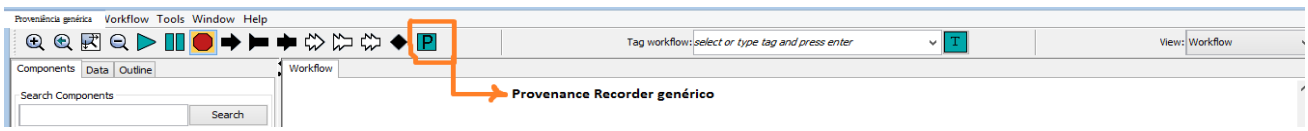

Figura 4. Exemplo do ator *Provenance Recorder* genérico do Kepler habilitado.

```
<?xml version="1.0"?>
<config>
  <provenance>
    <!--
      Default settings for provenance. Each name in a pair is the
      parameter name for the provenance recorder. (However some
      parameters only are used based on the output type).
    -->
    <defaultSettings>
      <!-- the output to use. See recordingTypes below for options. -->
      <pair>
        <name>Recording Type</name>
        <value>SQL-SPA-v8</value>
      </pair>
      <!-- host name running the database -->
      <pair>
        <name>DB Host</name>
        <value>localhost</value>
      </pair>
      <!-- database name (i.e., schema or sid) -->
      <pair>
        <name>DB Name</name>
```

Figura 5. Fragmento do arquivo de configuração do *Provenance Recorder* genérico.

Uma outra maneira de registrar a proveniência retrospectiva no Kepler se dá através da seleção do ator *Provenance Recorder*, que é o elemento do *workflow* responsável por configurar a proveniência de forma individualizada, ou seja, somente o *workflow* que está aberto no momento terá esta configuração. Ao clicar no ator *Provenance Recorder* será aberta uma tela de configuração (Figura 6), onde será possível definir parâmetros do tipo de SGBD (PostgreSQL, Oracle, MySQL e HSQL), nome do banco de dados, nome do usuário e senha, entre outros. Através desses parâmetros é possível fazer a conexão com o banco de dados escolhido. O nome do banco de dados que receberá a proveniência de dados deve ser o mesmo que está configurado no SGWfC Kepler ao ter sido previamente criado.

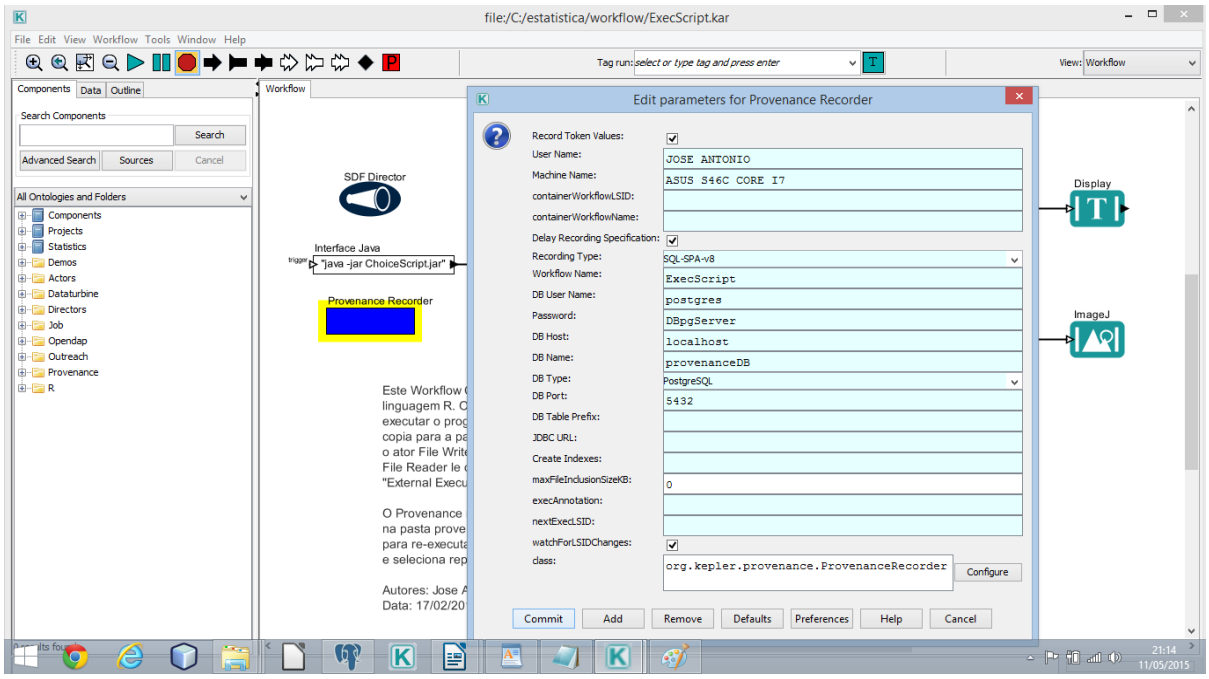

Figura 6. Configuração do ator *Provenance Recorder* do Kepler individualizado.

A opção padrão do "Recording Type" na configuração do PR é "SQL-SPA-v8", que habilita os tipos de bancos de dados citados acima, mas se selecionar a opção "OPM XML", a proveniência de dados será gravada em um arquivo XML no padrão OPM. A vantagem desse formato é que esse arquivo pode ser exportado para outro SGWfC que trabalhe com o padrão OPM. A desvantagem dessa opção é que não será possível gerar reexecuções do *workflow* através da funcionalidade do Kepler "Reporting" (Figura 7).

Para reexecutar o *workflow* deve-se acessar a opção "Reporting" na barra de comandos no canto superior direito da interface (Figura 7). Pode-se fazer busca do *workflow* a ser executado ou excluir um ou mais *workflows* da lista e automaticamente são excluídos do banco de dados definido na configuração do *Provenance Recorder*. A execução e reexecução de *workflows* no Kepler gera a proveniência retrospectiva e com isso a reprodutibilidade dos parâmetros, descritores de proveniência, etc.

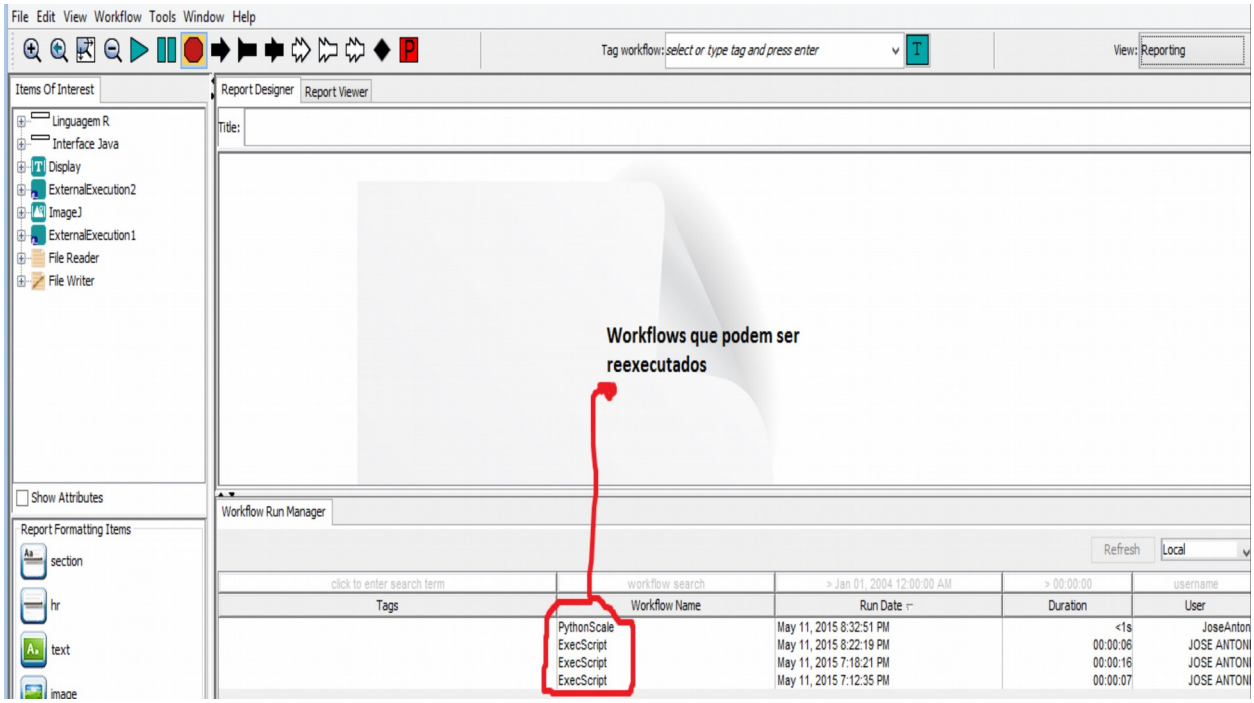

Figura 7. Tela de reexecução dos *workflows* no ambiente do Kepler.

3.2.3 SisGExp

O SisGExp é um aplicativo Web que utiliza a tecnologia Java EE e é o responsável pelo registro de todo o planejamento e acompanhamento dos dados experimentais coletados pelo pesquisador. O estilo arquitetural utilizado foi o MVC (modelo, visão e controle), pois permite a melhor separação das camadas: lógica (camadas *Core*), negócio (serviços de configuração) e apresentação (camada de interface). Foi utilizado o *framework* JavaServer Faces (JSF) em conjunto com a suíte PrimeFaces para implementar a interface e o controle da aplicação. O servidor de aplicação utilizado para suportar todas as especificações da API Java EE é o GlassFish 4.0.

As bibliotecas Primefaces 5.2, commons-fileupload-1.3.1.jar e commons-io-2.4.jar são utilizadas no desenvolvimento do SisGExp. A Primefaces possui inúmeros recursos (botões, barras, janelas, interfaces, entre outros) que permitem customizações mais apuradas para a camada *interface* com o usuário. As duas últimas são responsáveis pelas operações de *upload* e *download* de arquivos. O Capítulo 5 aborda o SisGExp na sua completude.

3.2.4 Meta-*Workflow* ExecScript

O meta-*workflow* ExecScript descrito na *Camada de serviços de Interface* da RFlow é o responsável pelo processamento dos *scripts* R. Ele representa uma solução genérica usada para encapsular *scripts* R legados. Ele atua como um *wrapper* que encapsula as funções estatísticas do *script* sob a forma de uma ou mais atividades de *workflow* concreto, permitindo sua execução em um SGWfC com todos os benefícios inerentes ao sistema, além da

funcionalidade da captura de proveniência. O meta-*workflow* permite a (re)utilização dos *scripts* R legados sob a forma de *workflows* científicos sem que seja necessário refatorar ou recodificá-los. A execução do meta-*workflow* é de responsabilidade do SGWfC, assim, se tira proveito de facilidades que estão ausentes no sistema R, tais como controle de execução, rastreabilidade, reprodutibilidade e a coleta de proveniência sobre cada execução. Será discutido no Capítulo 4.

3.3 Categorização dos Experimentos

Os experimentos utilizados nesta dissertação para avaliar a arquitetura RFlow são das seguintes áreas: Estatística, Agropecuária e Genética.

Estima-se que mais de 95% dos experimentos realizados na Embrapa Agrobiologia registram os dados do experimento em arquivos texto ou planilha de dados. No entanto, com o intuito de ampliar a gama dos testes da arquitetura RFlow, foram avaliados mais dois experimentos de domínios diferentes.

Os experimentos foram classificados em três categorias:

a) Categoria 1: *scripts* R que utilizam dados internos dispostos no próprio corpo do *script*;

b) Categoria 2: *scripts* R que utilizam dados externos armazenados localmente em arquivos;

c) Categoria 3: *scripts* R que utilizam dados externos remotamente armazenados na Web ou em outros servidores.

Por questões de ética e restrições de confidencialidade dos dados da Embrapa, nesta dissertação são utilizados dados sintéticos para os experimentos da categoria 2, que são os oriundos da Embrapa Agrobiologia. Os experimentos da categoria 1 (GEKKOQUANT, 2012) e categoria 3 (BIOINFORMATICS, 2015) são de domínio público e podem ser baixados diretamente da Internet.

3.3.1 Categoria 1: *Script* R que utiliza dados internos

Essa categoria de *script* R é possivelmente a mais simples. Os dados se encontram descritos no próprio corpo do *script*. O *script* R pode ser acessado na página http://gekkoquant.com/2012/05/26/neural-networks-with-r-simple-example/ e seu código-fonte está disponível no ANEXO A.

Resumidamente, esse *script* ilustra o funcionamento de uma rede neural *Perceptron Multi-Camadas (MLP),* treinada com o algoritmo *backpropagation.* Nesse exemplo é gerado randomicamente cinquenta números uniformemente distribuídos entre 0 e 100. A rede é treinada com dez neurônios ocultos para calcular a raiz quadrada dos cinquenta números gerados. Por fim, a rede é testada com dez números e impressa a imagem da rede neural e os resultados aproximados.

3.3.2 Categoria 2: *Script* R que utiliza dados externos armazenados localmente

Essa é uma das abordagens mais utilizadas na Embrapa, sendo foco deste trabalho para a coleta de descritores de proveniência em experimento agrícola. Geralmente, nesses experimentos são geradas planilhas de dados ou arquivos em formato *txt, csv, xls*, entre outros, referentes aos tratamentos aplicados às parcelas. Nesses arquivos estão contidas as variáveis independentes (fatores e seus níveis) e os dados coletados durante o acompanhamento do experimento (variáveis dependentes).

O *script* R utilizado como exemplo desta categoria se refere à análise de fertilidade do solo. Essas análises são de grande importância econômica e ecológica, pois é a partir das análises e seus resultados que serão feitas as recomendações técnicas da adubação/calagem nas áreas a serem cultivadas. Essa categoria de *scripts* utiliza massas de dados sob a forma de arquivos de entrada (o número de registros varia em função do tamanho da área avaliada) e emprega uma ampla variedade de funções estatísticas e gráficas do R (regressão linear, análise multivariada, variância, teste de normalidade, amostragem, entre outras). Os dados utilizados no estudo de caso são de natureza sintética e representam valores residuais de amostras de materiais orgânicos coletados em várias amostras de solos.

O código-fonte do *script,* denominado aqui de “scriptPesq.R” está disponível no ANEXO B. O *script* processa as funções estatísticas em um conjunto de dados que estão em um arquivo denominado “dadosPesq.txt”. O arquivo “dadosPesq.txt” é criado a partir da coleta de dados realizada durante o acompanhamento do experimento no campo.

Os resultados gerados por esse *script* são saídas textuais e gráficas.

3.3.3 Categoria 3: *Script* R que utiliza dados externos remotamente armazenados na Web ou servidores remotos

Esse *script* é de domínio da bioinformática. Ele utiliza uma sequência de genomas como entrada e compara com sequências de um arquivo remoto. O *script* R que foi testado está disponível na página http://manuals.bioinformatics.ucr.edu/home/ht-seq. Durante o processamento do *script* é acessado o arquivo de dados remoto que está localizado em http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/My_R_Scripts/AA.txt.

O arquivo AA.txt contém um conjunto de proteínas que será utilizado para ser comparado com a sequência de DNA criada randomicamente a partir da combinação ("ATGCAGACATAGTG", "ATGAACATAGATCC", "GTACAGATCAC"). Por fim, é impressa a sequência gerada e gravada em um arquivo no formato Fasta com o nome de myseq.fasta. O código-fonte do *script* está disponível no ANEXO C.

4 ARQUITETURA RFLOW

A reprodutibilidade de experimentos científicos *in silico* mediados por computador e a troca aberta de conhecimentos, dados e materiais entre times de pesquisa formam a espinha dorsal do progresso científico (CASADEVALL; FANG, 2010), (LI et al., 2011). No entanto, apesar de a reprodutibilidade ser uma das bases mais fundamentais da Ciência (POPPER, 1959), na área da ciência da computação existem muitas discussões e trabalhos científicos que mostram que ainda é necessário expor e garantir a reprodutibilidade dessa classe de experimentos (*in silico*). Existem até mesmo publicações que expõem o problema de modo aberto (PENG, 2011). Por exemplo, algumas revistas e congressos de primeira linha em computação (VLDB, SIGMOD, entre outros) exigem acesso aos executáveis ou aos conjuntos de dados.

A reprodutibilidade por si só não garante a qualidade de uma pesquisa ou a sua veracidade. No entanto, a reprodutibilidade de experimentos *in silico,* em teoria, deveria apresentar registros de *log* detalhados, descrição detalhada dos procedimentos, parâmetros utilizados, códigos de computador disponíveis para terceiros, dados de entrada e de saída, parâmetros, descrição de equipamentos utilizados, entre outros. Para alguns autores, a falta de um desses elementos classifica a pesquisa científica como falsa (LOANNIDIS, 2005).

Por exemplo, na área da Bioinformática, periódicos e artigos demandam a explicitação das técnicas de reprodutibilidade, correlacionando a publicação dos artigos com a execução dos *workflows* científicos e sua proveniência (GONZÁLEZ-BELTRÁN et al., 2015) (NASCIMENTO; CRUZ, 2015). Na área de computação, Freire et al. (2012) discutem especificamente o tema e também apontam como viáveis a realização dos experimentos *in silico* através de *workflows* científicos. Mais especificamente, na área de Bioestatística, periódicos tais como o *Jornal of Biostatistics* encoraja os autores cujos trabalhos foram aceitos a torná-los reproduzíveis por terceiros (PENG, 2011).

Neste capítulo é apresentado a arquitetura RFlow (NASCIMENTO; CRUZ, 2013), com vistas a contribuir com a possível redução dos problemas de baixa reprodutibilidade de *workflows* estatísticos no contexto de ambientes de pesquisas.

4.1 Arquitetura RFlow

A arquitetura RFlow é caracterizada, segundo os critérios classificatórios da taxonomia proposta por Cruz et al. (2009), como uma arquitetura de captura semiautomática de proveniência que utiliza mecanismos externos que atuam no nível do *workflow* por meio de técnicas de anotação. Nesse caso, os descritores de proveniência são capturados pelos sistemas de proveniência acoplados ao SGWfC que controlam a execução do *workflow* estatístico.

O benefício de trabalhar nesse nível reside no fato de que os *script* R legados são preservados, não necessitam ser adaptados ou (re)codificados para se beneficiarem de suporte do mecanismo de captura de proveniência do SGWfC.

A proveniência capturada é de natureza prospectiva e retrospectiva, de baixa granulosidade (detalhe do dado) e orientada a dados. É possível coletar descritores detalhados

sobre as execuções individualizadas de cada instância do *workflow*. Os descritores são armazenados de modo centralizado em um repositório do tipo relacional que está desacoplado do sistema de coleta. Essas características facilitam a atualização e substituição de módulos da arquitetura em atividades de manutenção corretiva ou adaptativa. A arquitetura RFlow é composta por diversas camadas e componentes.

A Figura 8 apresenta uma representação conceitual da arquitetura.

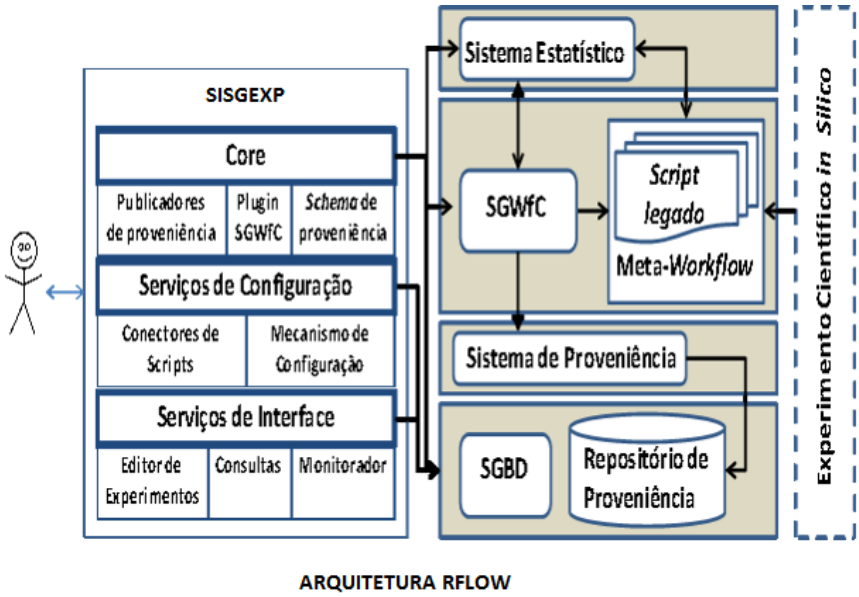

Figura 8. Arquitetura RFlow, camadas e componentes.

4.2 Camadas, Serviços e Componentes

A Figura 8 apresenta a arquitetura RFlow de acordo com o estilo múltiplas camadas (QIN; XING; ZHENG, 2008). A arquitetura é definida em três camadas principais:

(i) *Camada Core* - contém os componentes básicos para o funcionamento da arquitetura. O Core permite a inserção de *plugins* que são responsáveis pela configuração da tríade meta-*workflow*, SGWfC e repositório de proveniência. O componente publicador de proveniência prospectiva representa os serviços relacionados à publicação de dados de proveniência prospectiva. Analogamente, o componente publicador de proveniência retrospectiva representa os serviços relacionados à publicação de proveniência retrospectiva. Estes componentes são necessários devido às naturezas distintas da proveniência que operam no nível de *workflow* e que são coletadas inicialmente durante a configuração e posteriormente durante a execução dos *workflows*. O componente *schema* de proveniência define o modelo de dados compatível com as especificações de proveniências (OPM, PROV), que será utilizada pela abordagem para armazenar os descritores de proveniência.

(ii) *Camada de Serviços de configuração* - contém os componentes que são utilizados para viabilizar a configuração da RFlow. O componente mecanismo de

configuração é responsável por definir qual o SGWfC e o sistema de coleta de proveniência que são utilizados no experimento. Os componentes conectores de *scripts* são utilizados para facilitar o processo de carga dos *scripts*, conexão dos sistemas estatísticos e os meta-*workflows*.

(iii) *Camada de serviços de Interface* - oferecem ao pesquisador uma interface *Web* para controlar / executar seus experimentos. Utiliza componentes de editor de experimentos, permitindo que os pesquisadores cadastrem seus *scripts* R, meta-*workflows* e também descrições dos experimentos. Os serviços de interface, opcionalmente, poderão incluir atividades de monitoramento e de consulta de proveniência. As atividades de monitoramento permitem que os pesquisadores acompanhem a execução dos seus experimentos. As atividades de consulta realizam consultas sobre os descritores de proveniência capturados ao longo da configuração e execução do experimento. Além desses componentes, outros podem ser acrescentados na interface de acordo com a necessidade.

4.3 Meta-*Workflow* ExecScript

O meta-*workflow* ExecScript (Figura 9) é o *workflow* do tipo concreto responsável pela execução dos *workflows* estatísticos no SGWfC Kepler. Ele é composto por um sub-*workflow* e por atores genéricos do tipo "External Execution", que são os responsáveis por se comunicarem tanto com os componentes da *interface* da RFlow quanto com os demais atores do artefato ("Constant", "File writer" e "File Reader"). Os resultados gráficos são gerados pelos atores "Display" e "ImageJ". O "Display" mostra texto plano e o "ImageJ" apresenta gráficos mais sofisticados e imagens do tipo (tiff, jpeg, gif, etc).

O ExecScript permite o reúso dos *workflows* estatísticos sem a necessidade de alterar seus códigos-fonte. Isso é muito útil para os *scripts* legados ou nos casos do pesquisador que não conhece bem a linguagem R, pois como executa no ambiente do SGWfC, já provê o *script* de toda a infraestrutura do SGWfC, por exemplo, conectividade com banco de dados, comunicação web, gráficos, proveniência, etc.

O ExecScript foi configurado no Kepler para registrar os dados de proveniência no Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL. O componente utilizado para fazer a configuração de parâmetros de Banco de Dados é o *Provenance Recorder* (PR). O PR é o ator do Kepler que possui os recursos necessários para registrar a proveniência retrospectiva no PostgreSQL.

O ExecScript possui um sub-*workflow* representado pelo ator subExecScript (Figura 10), que é o responsável pela vinculação da proveniência prospectiva com a proveniência retrospectiva durante a execução do *workflow* estatístico (*script* R). A proveniência prospectiva que é realizada pelo SisGExp popula o banco de dados provenanceDB no esquema *expdados*. A tabela "wf_script" do esquema *expdados* é populada com dados de duas tabelas: o identificador (oid) da tabela "workflow_exec" do esquema *public* (Kepler), e com o oid da tabela "script" do esquema expdados (SisGExp). Isso é feito no final da execução do *workflow* estatístico pelo ExecScript. Dessa maneira, tem-se a amarração entre as duas tabelas dos dois bancos de dados. Com isso é possível saber:

(i) qual *script* foi executado, quem executou, a data de execução;

(ii) status da execução, se houve erro, qual erro, se foi execução parcial;

(iii) o *script* executado pertence a qual experimento, a qual publicação, entre outras.

Vale ressaltar, que após a configuração do ator *Provenance Recorder*, todo o processo para a captura da proveniência retrospectiva, amarração com a proveniência prospectiva e por fim as consultas são todas realizadas automaticamente, ou seja, não há intervenção manual para ajustes ou algo parecido.

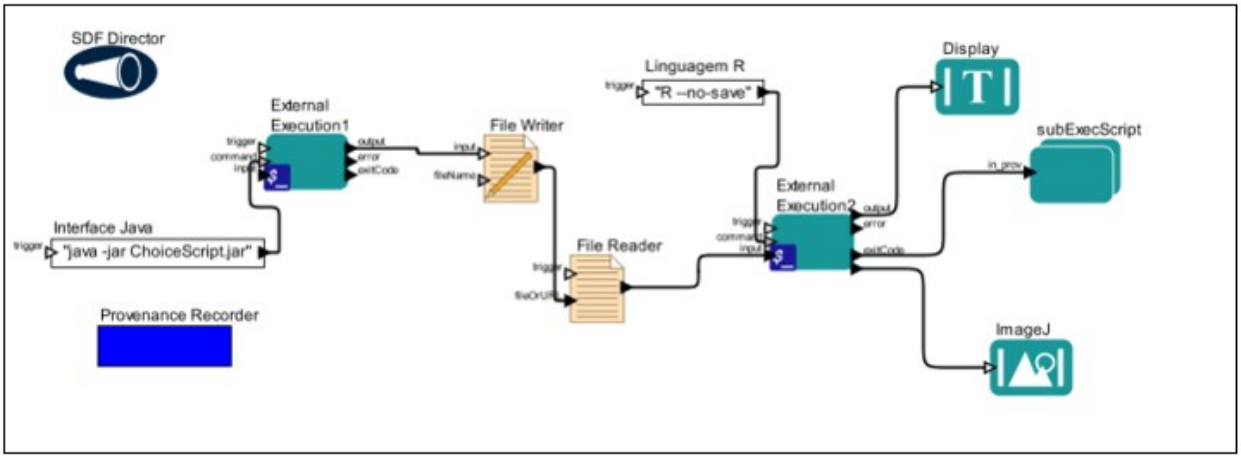

Figura 9. Meta-*workflow* ExecScript codificado para o SGWfC Kepler.

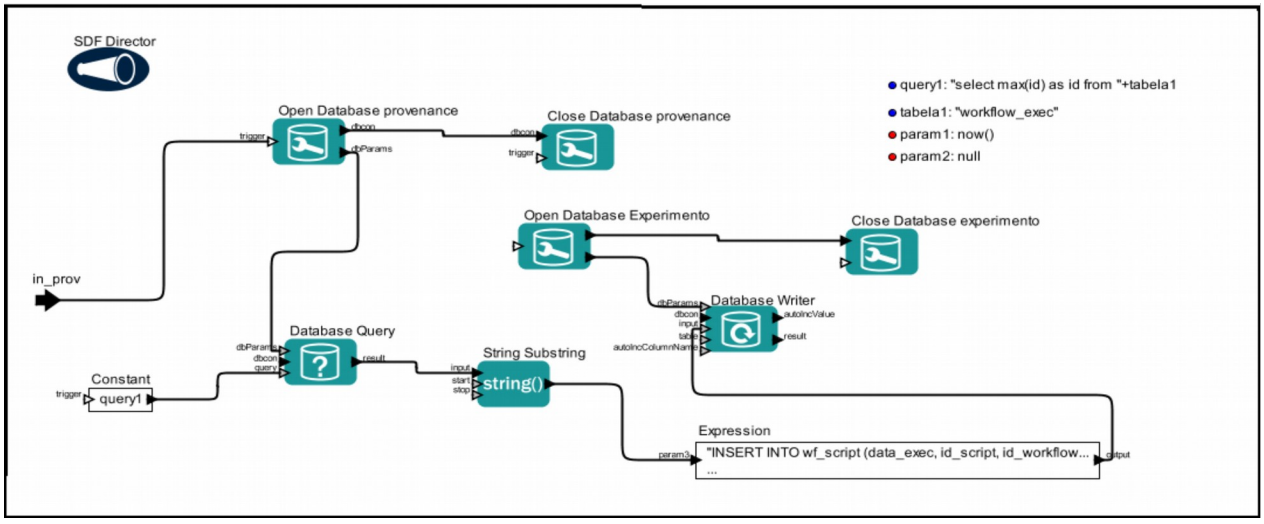

Figura 10. SubWorkflow do ExecScript representado pelo ator subExecScript.

4.4 Bancos de Dados provenanceDB

O BD provenanceDB é formado por dois esquemas: *public* e expdados. Essa abordagem simplifica o processo de coleta de proveniência prospectiva e retrospectiva, uma vez que será manipulado somente um banco de dados no meta-*workflow* ExecScript. Vale lembrar que esquemas no PostgreSQL são bancos de dados, ou seja, em vez de criar dois bancos de dados independentes são criados dois esquemas em um único banco de dados.

4.4.1 Esquema Public: modelo de dados do Kepler

O esquema *public* está relacionado com a proveniência retrospectiva, ou seja, utiliza o modelo de dados do Kepler (Figura 11). Esse modelo é formado por 17 tabelas, onde são inseridos os dados e metadados provenientes da execução de um *workflow* científico. Essas tabelas são criadas automaticamente quando da instalação e configuração do ator (componente) *Provenance Recorder* no SGWfC Kepler.

A proveniência retrospectiva é gerada pela execução de cada instância de um *workflow* concreto no Kepler, e é representada por três tipos de informação: o conteúdo ou especificação de *workflows*, como essas especificações mudam ao longo do tempo e os eventos que ocorrem durante a execução do *workflow* no Kepler. Para cada instância de *workflow* executada no SGWfC Kepler é gerado um identificador único.

O armazenamento da proveniência é local e centralizado. O metamodelo é OPM-compatível e suas relações mapeiam os principais componentes dos *workflows* concretos (atores, diretores, portas, parâmetros e o contexto de execução). Os descritores de proveniência retrospectiva podem ser consultados através de simples consultas SQL (Structured Query Language) contidos nos serviços de interface.

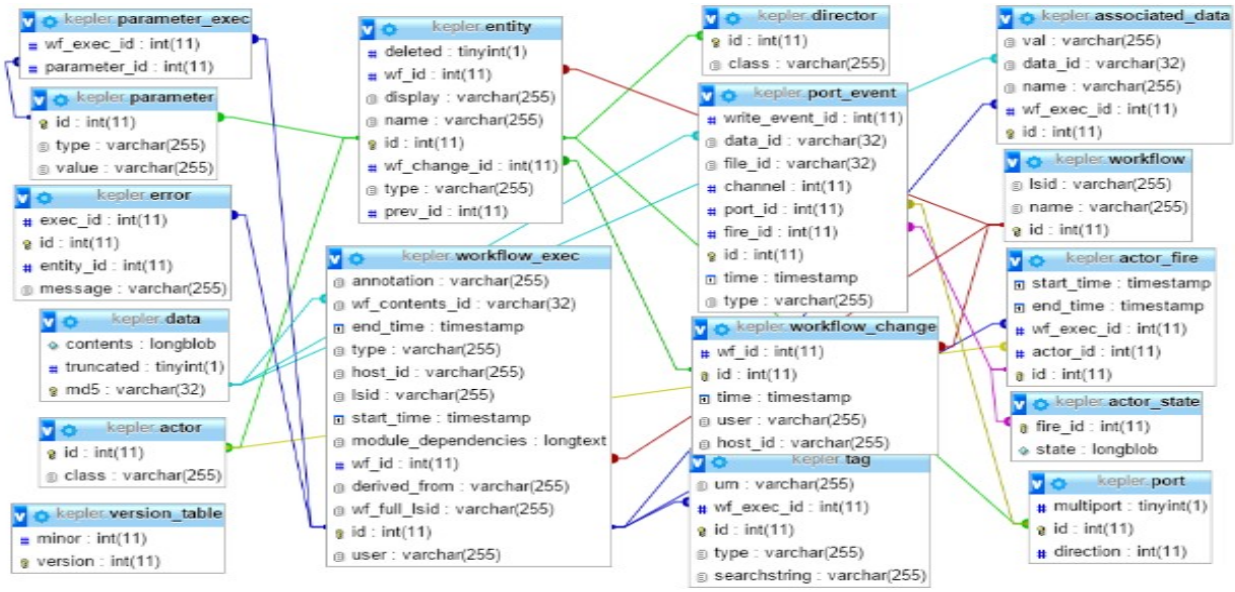

Figura 11. Modelo de dados (Esquema *public*) do SGWfC Kepler (Prov. Retrospectiva). (ALTINTAS et al., 2006).

4.4.2 Esquema Expdados: modelo de dados do SisGExp

O esquema expdados (Figura 12) foi concebido e implementado para atender uma demanda antiga da Embrapa Agrobiologia. Um banco de dados de experimentos agrícolas que registre o ciclo de vida de um experimento.

O esquema é formado por 11 tabelas e seu modelo lógico é compatível com uma variação baseada no metamodelo OPM. Manter a compatibilidade com o modelo do Kepler facilita a interoperabilidade dos dados e eventos ocorridos durante a execução do experimento.

No esquema expdados são registrados os dados do planejamento e acompanhamento dos experimentos científicos, ou seja, a proveniência prospectiva. A tabela "wf_script" é a responsável pela vinculação entre os dados da proveniência retrospectiva (esquema public) e os dados da proveniência prospectiva (esquema expdados) (Esquema está no ANEXO D).

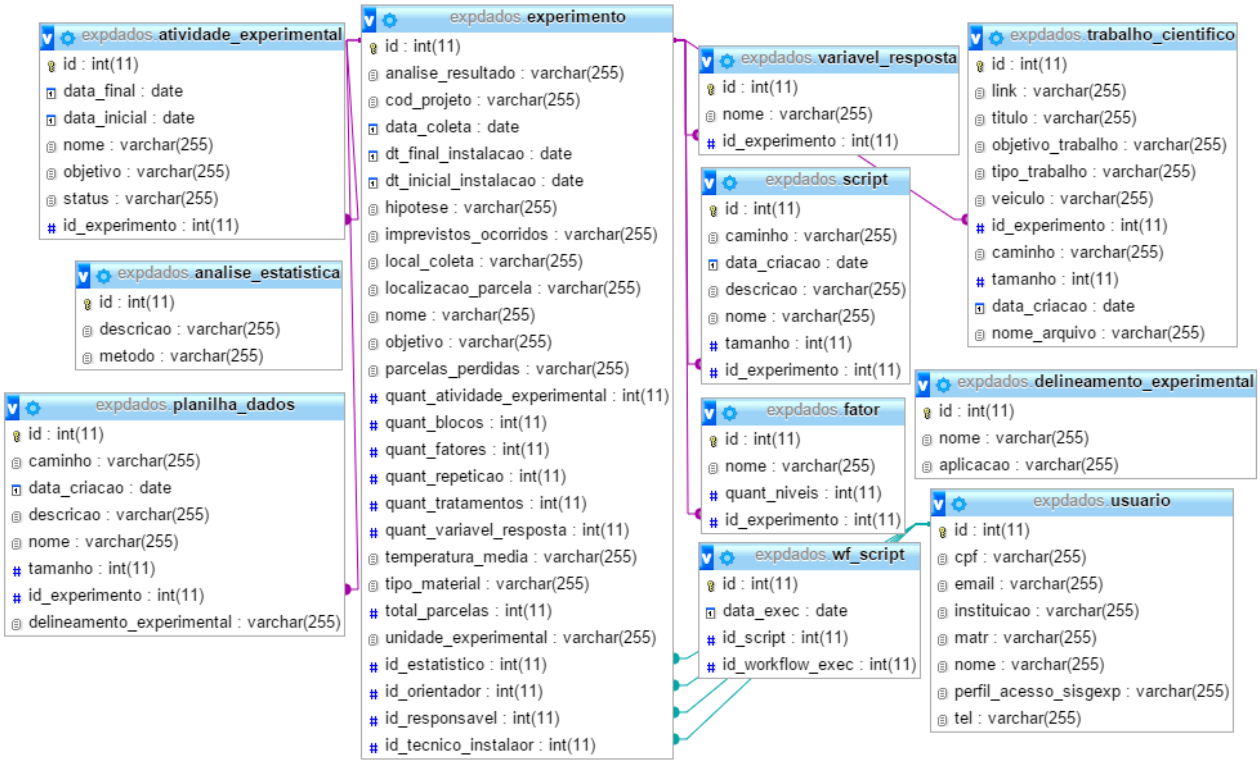

Figura 12. Modelo de dados (Esquema *expdados)* do SisGExp (Prov. Prospectiva).

4.5 Interação da RFlow com seus componentes

As três camadas da arquitetura RFlow (*core*, configuração e interface) interagem com o SisGExp em fases distintas: planejamento do experimento, configuração das tabelas auxiliares, execução do experimento e consultas.

O aplicativo SisGExp é utilizado para registrar a proveniência prospectiva de cada experimento gerenciado pelo pesquisador; invocar o SGWfC Kepler para executar os *scripts* R legados e disponibilizar os dados experimentais e os processos científicos que os manipulam de maneira organizada e *online*. O SisGExp foi concebido e implementado para

ser parte da arquitetura RFlow.

A arquitetura RFlow atua como uma plataforma composta por um conjunto de componentes integrados, em que cada componente tem um propósito bem definido. Resumidamente, os quatro principais componentes que integram a arquitetura são: Sistema de Gestão de Experimentos (SisGExp), Linguagem R, SGWfC Kepler e o Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL.

A Figura 13 apresenta uma visão conceitual e simplificada dos fluxos de controle da arquitetura, seus controles de leitura (R) e escrita (W) e dos componentes responsáveis por viabilizar a reprodutibilidade dos experimentos científicos apoiados por *workflows* estatísticos, bem como a coleta e armazenamento da proveniência e resultados.

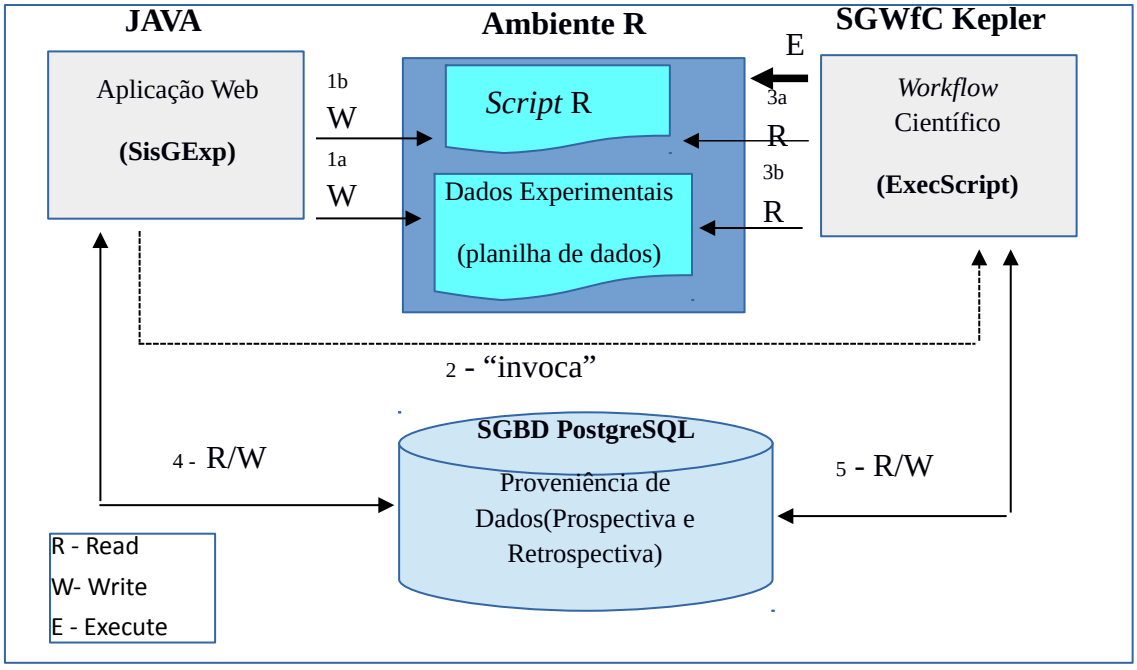

Figura 13. Visão simplificada do fluxo de controles do SisGExp e o Meta-*workflow* na arquitetura RFlow.

Conceitualmente, a arquitetura RFlow permite que o pesquisador configure os dados, parâmetros e descritores do seu experimento através do SisGExp (1a e 1b). A seguir (2) ele invoca de maneira transparente e remotamente o SGWfC *Kepler* que parametriza automaticamente (3a e 3b) um meta-*workflow* genérico (denominado ExecScript) que encapsula e controla a execução dos *scripts* R legados no ambiente R (selecionado em 1b).

O SGWfC *Kepler* orquestra a execução do meta-*workflow* (3a e 5) e coleta a proveniência da execução do experimento baseado em meta-*workflow* que encapsula os *scripts* R de qualquer tipo através do serviço *Provenance Colector*. O SisGExp permite que o pesquisador monitore (4) remotamente a execução do experimento (5).

5 SISTEMA SISGEXP

O SisGExp é o módulo responsável pelo registro e acompanhamento dos experimentos agrícolas, bem como pelas consultas dos resultados estatísticos gerados pela execução dos *scripts* R legados.

5.1 Parametrização do SisGExp

O SisGExp é multiplataforma, capaz de executar nos sistemas operacionais *Linux* e *Windows*. Pode executar de modo local ou remotamente, uma vez que é um aplicativo *Web*. O SisGExp executa diversas operações de leitura e gravação de arquivos. Portanto, é necessário criar uma estrutura de pastas no sistema de arquivos do sistema operacional para possibilitar a leitura e gravação de arquivos de forma organizada e coordenada (Quadro 4).

Quadro 4. Estrutura de pastas utilizadas pela arquitetura RFlow

windows	Linux	Função
C:\experimentoRFlow	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow	Pasta principal.
C:\experimentoRFlow\interface	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow/interface	Pasta do programa Java desktop ChoiceScript. Usado somente se for executar localmente.
C:\experimentoRFlow\output	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow/output	Local de sincronização do script selecionado para gerar os resultados estatísticos.
C:\experimentoRFlow\upload	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow/upload	Pasta que armazenará as subpastas de dados, *scripts* e publicações anexadas pelo usuário.
C:\experimentoRFlow\upload\dados	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow/upload/dados	Arquivos de dados usados pelos *scripts* R.
C:\experimentoRFlow\upload\script	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow/upload/script	*Scripts* R que foram anexados.
C:\experimentoRFlow\upload\trabalhoCientifico	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow/trabalhoCientifico	Trabalhos científicos anexados.
C:\experimentoRFlow\workflow	usr/share/glassfish/glassfish/domains/domain1/applications/experimentoRFlow/workflow	Local do meta-*workflow* ExecScript.

5.2 Visão Geral do SisGExp

A Figura 14 mostra a tela de acesso ao sistema SisGExp. A princípio, o sistema executará experimentalmente na Embrapa Agrobiologia. Dessa forma, o acesso dar-se-á

através da matrícula do funcionário e uma senha interna que fica registrada nos servidores da Embrapa. Ao executar o login, o sistema identifica o usuário.

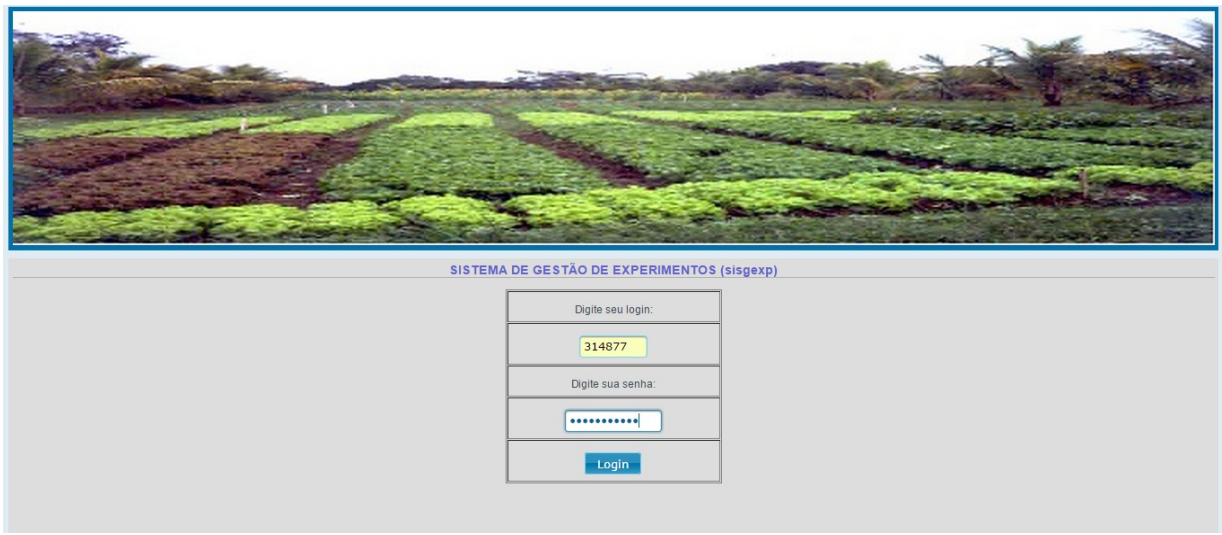

Figura 14. Tela de acesso ao SisGExp.

A Figura 15 exibe a tela principal do sistema e suas funcionalidades. É possível verificar que as principais funções estão representadas através de uma barra de menus.

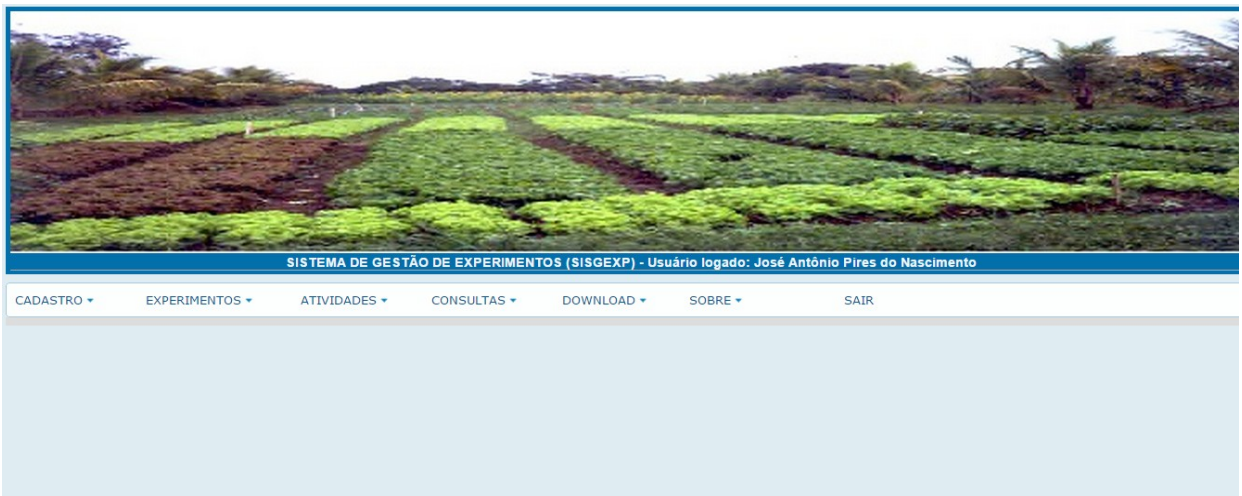

Figura 15. Tela principal do sistema.

Com relação à camada de persistência de dados e de proveniência, o SisGExp persiste os descritores de proveniência prospectiva e retrospectiva no SGBD PostgreSQL (Versão 9.3.4). O banco de dados é o provenanceDB, que foi segmentado em dois esquemas: *public* e *expdados*.

O SGWfC utilizado para executar o meta-*workflow* ExecScript é o Kepler, devido a sua estabilidade e, principalmente, a facilidade de integração com o sistema R através da utilização de atores genéricos.

A Figura 16 apresenta a correlação entre o ciclo de vida de um experimento agrícola com cada funcionalidade do SisGExp. Na coluna da esquerda estão as etapas correspondentes a cada fase de um ciclo experimental agrícola (planejamento, acompanhamento, resultado e análise). Na coluna da direta está relacionada sua operacionalização no sistema. Nas etapas de planejamento e acompanhamento são coletados dados de proveniência prospectiva.

Por exemplo, a etapa de *planejamento do experimento agrícola* é representada no SisGExp pela funcionalidade denominada *novo experimento*. A *etapa de acompanhamento do experimento (coleta de dados experimentais)* é representada no SisGExp através das funcionalidades *novas atividades* e a *planilha de dados*. A etapa de *obtenção de resultados estatísticos* corresponde à execução do experimento (meta-*workflow* e *script* R legado) propriamente dito nos ambientes computacionais R e Kepler e corresponde à coleta de dados de proveniência retrospectiva.

Por fim, a etapa de *Análises e Publicação dos Resultados Estatísticos* corresponde à publicação dos experimentos, dados e parâmetros utilizados e publicação dos seus resultados. Ela corresponde à publicação de dados de proveniência retrospectiva e prospectiva coletados durante o ciclo de vida do experimento agrícola.

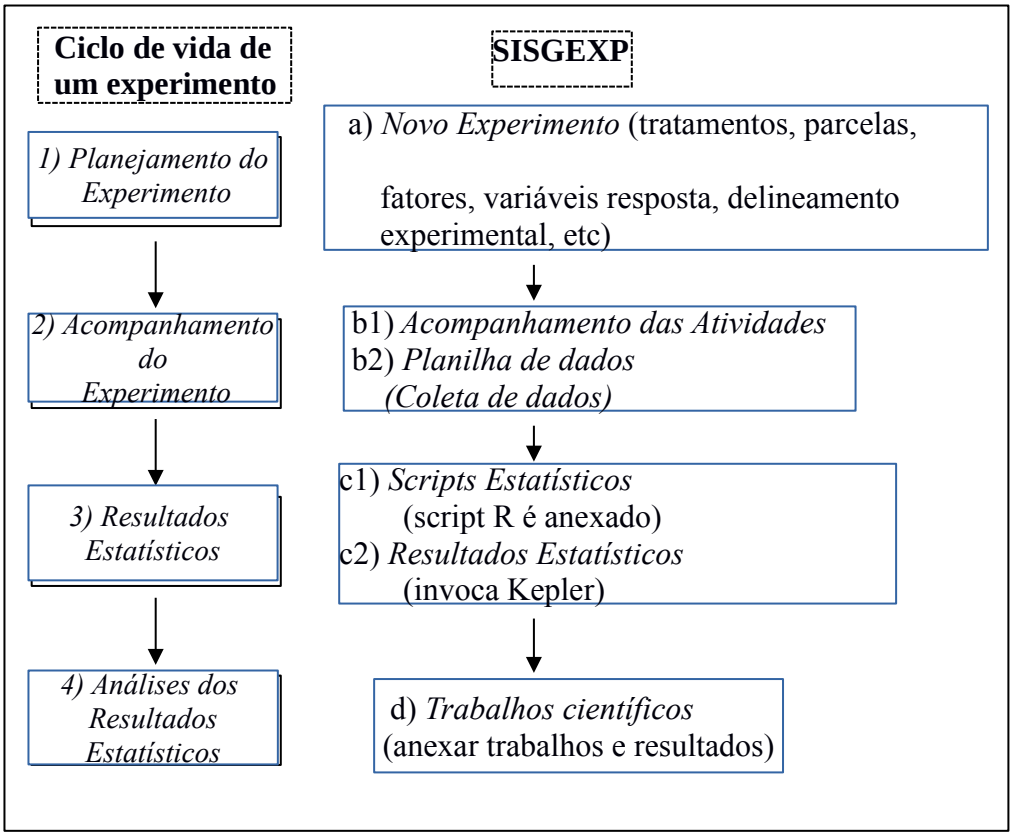

Figura 16. Correlação do ciclo de vida de um experimento agrícola e os passos no SisGExp.

Conforme anunciado, o SisGExp é o componente da arquitetura RFlow responsável pela captura da proveniência prospectiva, mais precisamente pela composição no âmbito do ciclo de vida de um experimento científico agrícola. Também age como uma ligação entre o ExecScript e a proveniência retrospectiva através das execuções individualizadas dos *scripts R* legados anexados pelos pesquisadores.

Nas próximas ilustrações são mostradas as principais telas do SisGExp para cada fase do ciclo de vida do experimento, conforme o estabelecido na Figura 16.

1) Planejamento do Experimento

Esta etapa corresponde a *Novo Experimento* no SisGExp. A Figura 17 exibe a tela em que o pesquisador faz o planejamento de seu experimento. São definidos os dados e parâmetros experimentais, por exemplo: nome do experimento, unidade experimental, local da coleta, objetivo do experimento, data inicial e final de instalação do experimento, fatores, variáveis resposta, delineamento experimental, responsável pelo experimento, entre outros. Esses dados e metadados representam a proveniência prospectiva do experimento científico.

Cada novo experimento recebe um identificador do objeto (*oid*). Esse código é único e identificará todas as atividades referentes a esse experimento. Esse identificador de experimento é vinculado com o identificador dos dados do Kepler durante a execução do meta-*workflow* ExecScript.

Figura 17. Tela de planejamento de um novo experimento agrícola.

2) Acompanhamento do Experimento

Esta etapa do ciclo de vida do experimento agrícola foi segmentada em duas funcionalidades no SisGExp: (b1) *novas atividades* e (b2) *planilha de dados*. A Figura 18 exibe a tela que corresponde a *novas atividades* e a Figura 19 a tela que corresponde a *planilha de dados*. Nessa fase são coletados os dados experimentais que podem estar associados a mais de uma atividade e em períodos diferentes. Após a finalização do experimento, a planilha de dados gerada é anexada ao SisGExp. Nessa planilha estão todos os valores quantitativos e qualitativos que o pesquisador observou nos tratamentos aplicados ao experimento. Os formatos de planilha de dados permitidos pelo SisGExp são: csv, txt, rtf, doc, docx, xls, xlsx, ods, odt, dbf. Essa planilha é armazenada no servidor e estará relacionada com o identificador do experimento. Corresponde a proveniência retrospectiva do experimento.

A planilha de dados é opcional, pois o SisGExp permite que haja experimento sem arquivos anexados. São os casos de experimentos que acessam dados remotos ou *scripts* com dados no próprio corpo do *script.*

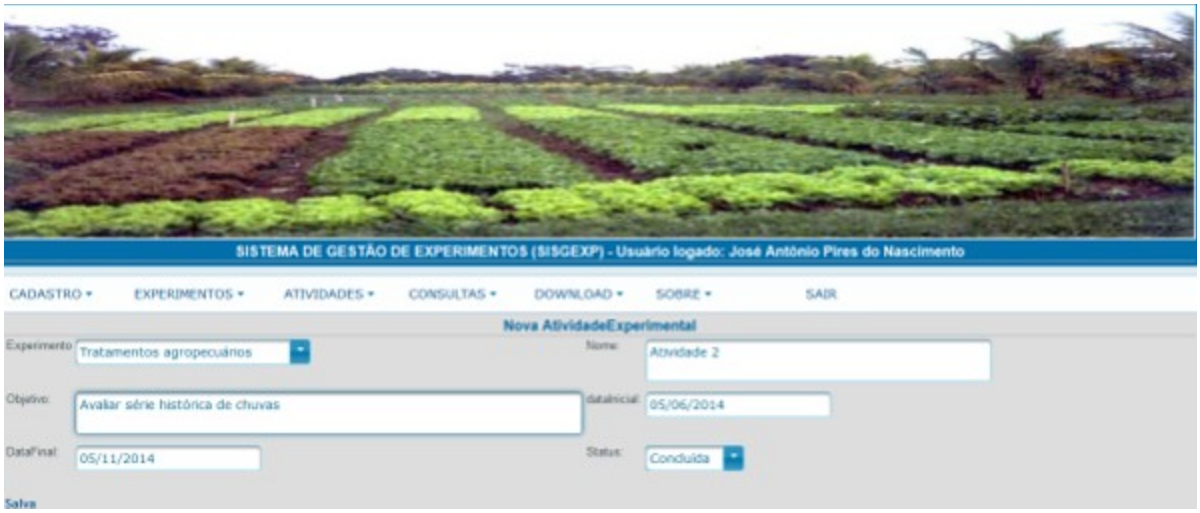

Figura 18. Novas atividades de um experimento agrícola.

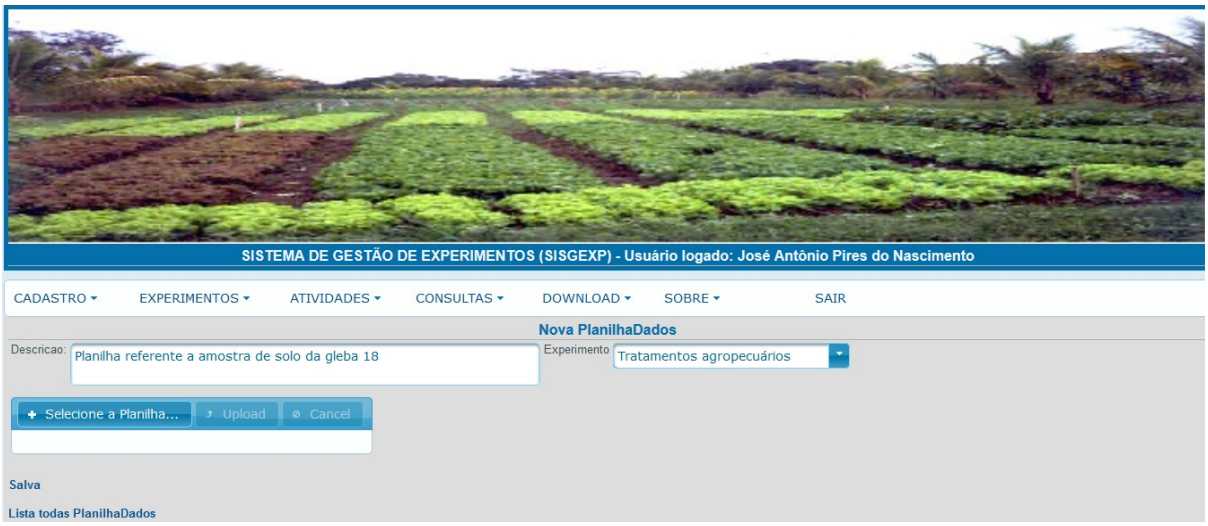

Figura 19. Tela que possibilita o pesquisador anexar a planilha com dados brutos coletados durante o acompanhamento do experimento.

3) Resultados Estatísticos

Esta etapa foi segmentada em duas funcionalidades no SisGExp: (c1) *Scripts Estatísticos* (*script* R é anexado) e (c2) *Resultados Estatísticos* (invoca o Kepler). A Figura 20 exibe a tela que permite ao profissional da área de estatística anexe o *script* R legado. Esse *script* representa as análises estatísticas realizadas pelo pesquisador ou profissional de estatística. As análises são baseadas no planejamento do experimento e nos dados coletados durante a fase de acompanhamento do experimento. Esses scripts possuem os seguintes formatos: txt, R, rtf, doc, docx, odt.

A Figura 21 exibe a tela com os *scripts* prontos para gerarem os Resultados Estatísticos *online*. Operacionalmente, a partir desse ponto, o SisGExp associará o meta-*workflow* ExecScript através desse *script* anexado pela interface. A proveniência retrospectiva é gerada nessa fase, onde são populadas as tabelas do esquema public (Kepler) e do esquema expdados (SisGExp), ambas do banco de dados provenanceDB.

Aqui se aplica o mesmo caso da planilha de dados, ou seja, é opcional anexar o *script*. Nessa situação, o SisGExp agirá como um repositório de proveniência prospectiva (repositório de publicações vinculadas aos experimentos).

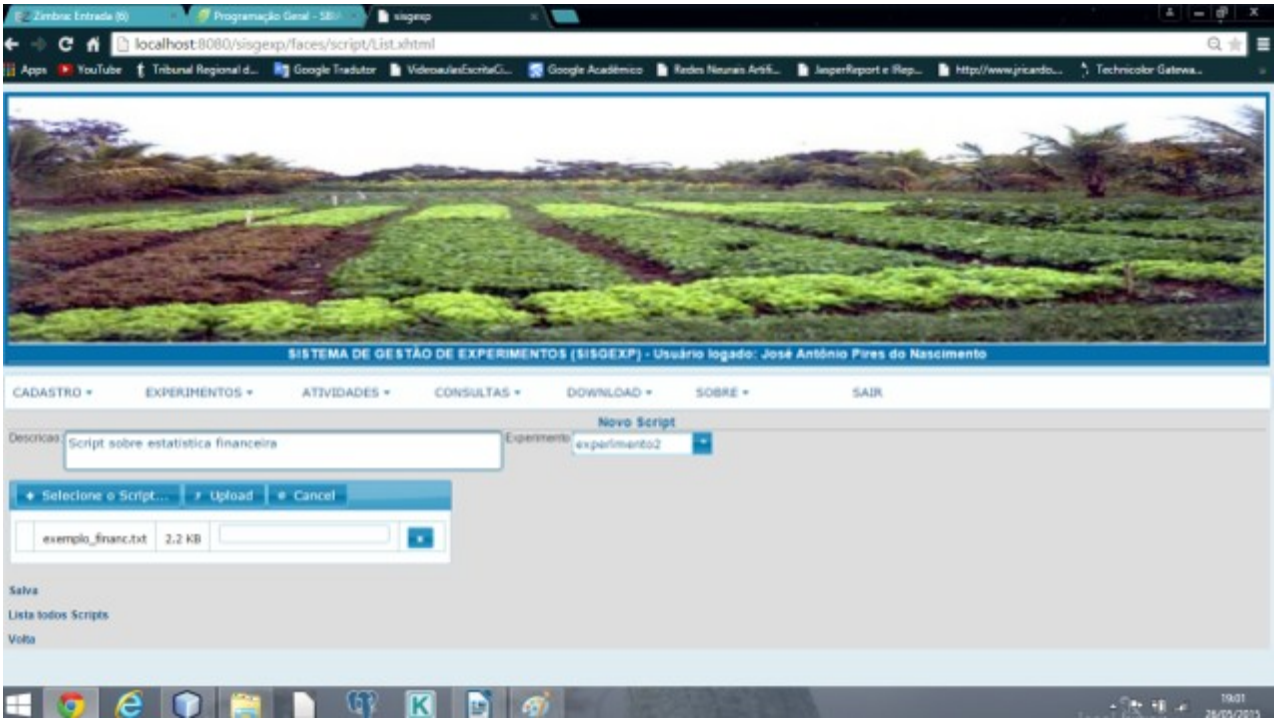

Figura 20. Tela de incorporação do *script* R legado com as análises estatísticas a serem aplicadas nos dados brutos coletados.

Figura 21. Tela de invocação do SGWfC Kepler e do meta-*workflow* ExecScript para executar o *script* R anexado.

4) Análises dos Resultados Estatísticos

Esta etapa corresponde ao término do ciclo de vida do experimento, uma vez que são anexadas as análises dos resultados experimentais. A Figura 22 exibe a tela em que são inseridos os trabalhos científicos que, obrigatoriamente, devem estar associados a um experimento. Na funcionalidade *Download* (Figura 23) é possível baixar os textos científicos referentes a artigos, dissertações, teses, entre outros.

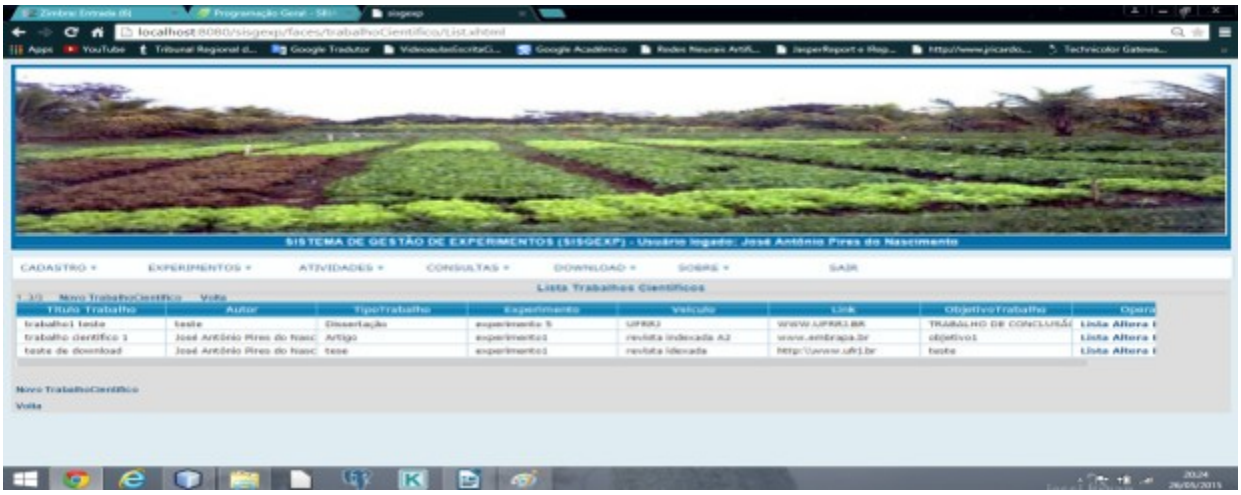

Figura 22. Tela de Exibição dos trabalhos científicos relacionados ao experimento.

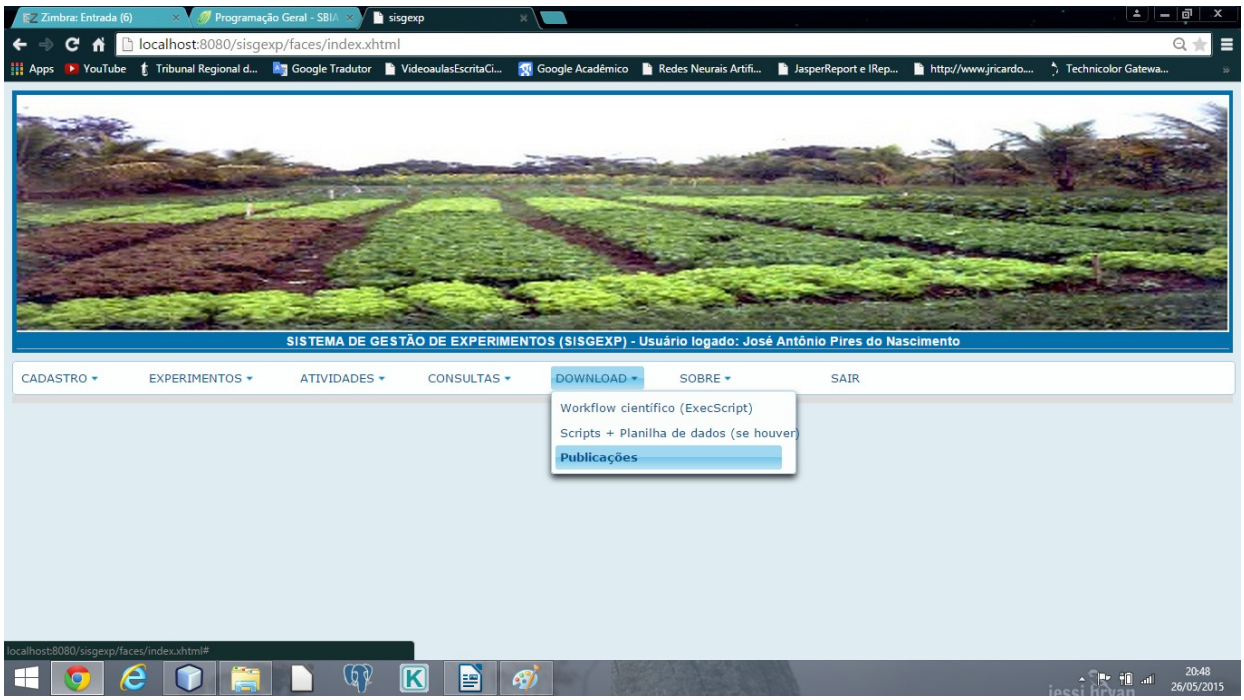

Figura 23. Tela de Downloads (*workflow*, *scripts*, planilhas, publicações).

6 AVALIAÇÃO EXPERIMENTAL DA ARQUITETURA

Para confirmar ou refutar a hipótese sobre a questão de pesquisa elaborada no Capítulo 1 é necessário verificar se a arquitetura RFlow atende aos requisitos propostos.

Com o intuito de avaliar qualitativamente a arquitetura RFlow através do SisGExp são abordados neste Capítulo os seguintes tópicos nas seções seguintes: a seção 6.1 descreve o ambiente de avaliação da arquitetura. Na seção 6.2 são avaliados experimentalmente as três categorias de *scripts* R que foram definidas no capítulo 3. Na seção 6.3 são realizadas consultas no banco de dados provenanceDB utilizando a linguagem SQL. Essas consultas são realizadas nos dados coletados durante a execução do SisGExp para os três experimentos avaliados. Na seção 6.4 são discutidos os principais resultados obtidos através da arquitetura RFlow, e por fim, na seção 6.5 são feitas algumas considerações.

Para maior clareza dos experimentos realizados, foram consideradas as seguintes premissas: (i) nas subseções 6.2.1, 6.2.2 e 6.2.3 são exibidos os relatórios em formato pdf ou txt relacionados ao processamento dos *scripts* R legado para cada categoria de experimento definida na seção de Materiais e Métodos; (ii) durante o processamento dos *scripts* R é gerada e coletada a proveniência retrospectiva; (iii) as três consultas SQL realizadas são: duas consultas relacionadas com a coleta de proveniências (prospectiva e retrospectiva), e, por fim, uma única consulta que aborda a correlação entre os dois tipos de proveniências associadas a um mesmo experimento.

6.1 Ambiente de Execução da RFlow

Todos os experimentos foram executados em um equipamento do tipo notebook com 8 GB de RAM e processador Intel *Core* i7 com conectividade na Internet através de uma placa de redes de 100Mpbs.

A arquitetura foi avaliada nos dois sistemas operacionais: Ubuntu 14.04 LTS e Windows 7.0. Foi executado nos dois navegadores de páginas: Firefox Versão 40.0.3 e Google Chrome Versão 45.0.2454.85. Não houve diferenças nos ambientes descritos.

6.2 Avaliação Qualitativa da RFlow

Para avaliar experimentalmente a arquitetura RFlow são consideradas as três categorias de *scripts* R, a saber:

a) Categoria 1: o *script* utiliza dados internos dispostos no próprio *script;*

b) Categoria 2: o *script* utiliza dados externos presentes em uma planilha de dados armazenada localmente;

c) Categoria 3: o *script* utiliza dados externos remotamente armazenados na Web ou em outros servidores.

A Figura 24 exibe a tela do SisGExp com as três categorias de experimentos de testes cadastrados. Esses experimentos foram definidos no Capítulo de Materiais e Métodos. Como já citado, para gerar os resultados estatísticos e registrar a proveniência retrospectiva é necessário que o *script* R legado esteja vinculado a cada um dos experimentos. Após o

cadastro dos experimentos, o estatístico ou o próprio pesquisador pode anexar o *script* R contendo as análises realizadas (Figura 25). Esse procedimento corresponde às fases de planejamento e acompanhamento do experimento científico na área agrícola.

Figura 24. Tela com as três categorias de experimentos de teste. Esses dados foram coletados na fase de planejamento e acompanhamento do experimento.

Figura 25. Tela com a lista dos *scripts* R anexados. Estão relacionados a cada categoria de experimento definida no Capítulo 3.

6.2.1 Experimentos da Categoria 1

É a categoria de *script* R mais simples, os dados se encontram no próprio corpo do *script*. Nesse experimento de teste uma rede neural é testada com dez números de entrada e como saída de dados é produzida a imagem da rede, exposto na Figura 26 e os resultados aproximados no Quadro 5.

Quadro 5. Resultado da execução do *script* R de categoria 1

Input	Expected Output	Neural Net Output
1	1	0.9984154698
4	2	2.0016716739
9	3	2.9982157822
16	4	3.9987940986
25	5	4.9949359964
36	6	6.0078252585

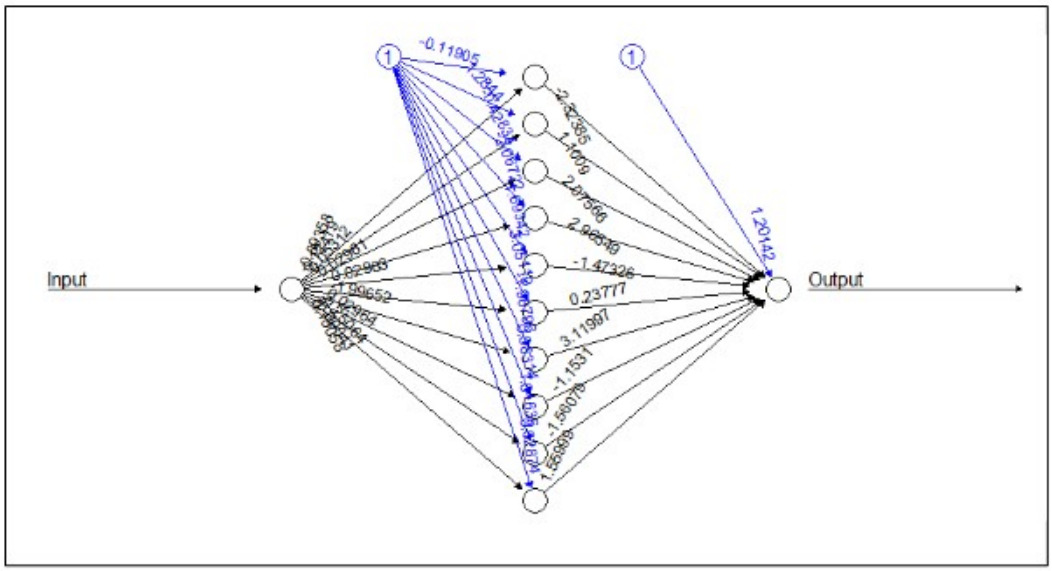

Figura 26. Rede neural com dez neurônios ocultos para cálculo da raiz quadrada.

6.2.2 Experimentos da Categoria 2

Como citado no Capítulo 3, essa categoria de experimento é a abordagem mais utilizada na Embrapa Agrobiologia. Geralmente nesses experimentos são geradas planilhas de dados ou arquivos em formato txt, csv, xls, etc, referentes aos tratamentos aplicados às

parcelas. Nesse caso são gerados vários gráficos. Para efeito de ilustração são mostrados dois deles (Figuras 27 e 28).

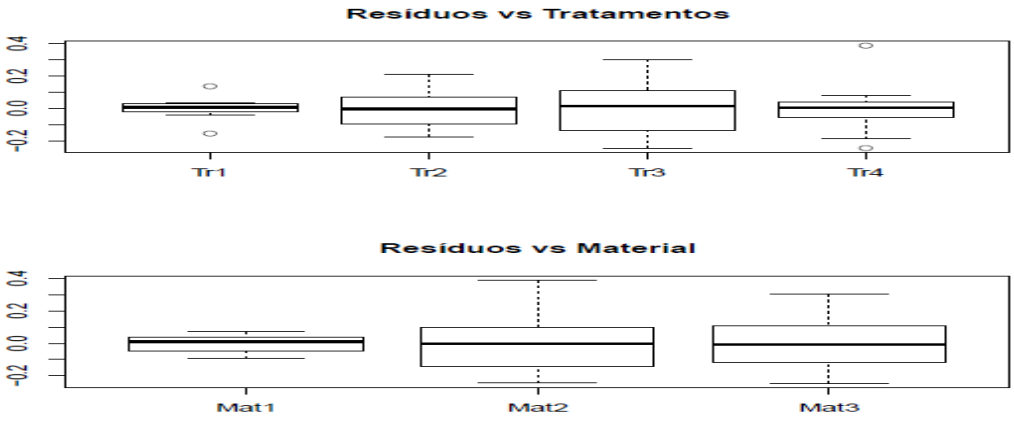

Figura 27. Relação entre Resíduos x Tratamentos e Resíduos x Material.

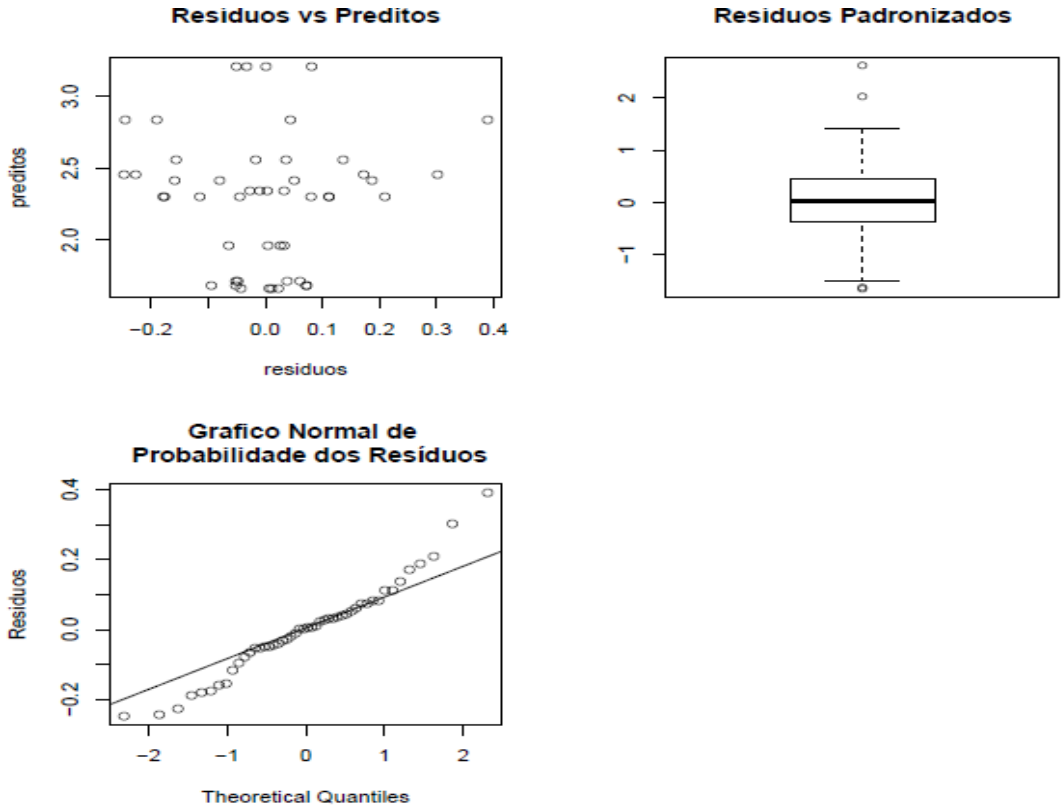

Figura 28. Gráfico Normal de Probabilidade dos Resíduos.

6.2.3 Experimentos da Categoria 3

O exemplo desta categoria é um *script* aplicado na área de bioinformática. Utiliza uma sequência de entrada de um genoma e compara com sequências de um arquivo remoto. Como resultado será impressa uma nova sequência e gravado um arquivo no formato Fasta com o nome de myseq.fasta (Figuras 29 e 30).

```
> ##################################################
> myseq <- sapply(1:12, function(x) paste(sample(c("A","T","G","C"), 40, replace=T), collapse=""))
> writeLines(myseq) # Prints the sequences to the screen, one per line.
CCCAGCCCCCGGCTTCAAATGGACGGAGGGACAGATCCCG
GCTACAGGGCAGTATGTGGGCACGACGAGGGAATATAGCG
GATACCCACCCACATTATTACTAGAACGGGCGGGTAAGCT
AACAACCTTGGTCGTCAGCGAGCGTCCCGCCAGAAGGATT
ATGGCTTCCTGTCTGTTCGCTCATCAGCAGCTCTGCACTG
AATTAGTAACCCCTCCGCCTGGTTAAGTCAGACACTCGTG
GTCAAACTGTGGAACAGCTAGCGAGGTTAACGAGGAGGCC
AGAGGTCCATAAGCCGTAACACGTTGTCCGTCTAGCACCG
TTTTTGGATCCGATGTCGGCTGTCGTGCTGGGGCGACCTA
AGAATGTGGAGTGCACGAATCGTTGAGCGTCAATGGAGTT
CGACCAACAATCCCAATTACGGTAACAAACAAGTCGGGGT
AGTCCGTCCAGTTAAGACTACGGTCTTACGAGACGCCAAC
> writeLines(strtrim(myseq, 10)) # The strtrim() function allows to print only the first section of the sequences.
CCCAGCCCCC
GCTACAGGGC
GATACCCACC
AACAACCTTG
ATGGCTTCCT
AATTAGTAAC
GTCAAACTGT
AGAGGTCCAT
TTTTTGGATC
AGAATGTGGA
CGACCAACAA
AGTCCGTCCA
> writeLines(strwrap(myseq, indent = 20)) # The strwrap() function provides wrapping, indenting and prefixing utilities.
                    CCCAGCCCCCGGCTTCAAATGGACGGAGGGACAGATCCCG
                    GCTACAGGGCAGTATGTGGGCACGACGAGGGAATATAGCG
                    GATACCCACCCACATTATTACTAGAACGGGCGGGTAAGCT
                    AACAACCTTGGTCGTCAGCGAGCGTCCCGCCAGAAGGATT
                    ATGGCTTCCTGTCTGTTCGCTCATCAGCAGCTCTGCACTG
                    AATTAGTAACCCCTCCGCCTGGTTAAGTCAGACACTCGTG
                    GTCAAACTGTGGAACAGCTAGCGAGGTTAACGAGGAGGCC
                    AGAGGTCCATAAGCCGTAACACGTTGTCCGTCTAGCACCG
                    TTTTTGGATCCGATGTCGGCTGTCGTGCTGGGGCGACCTA
                    AGAATGTGGAGTGCACGAATCGTTGAGCGTCAATGGAGTT
                    CGACCAACAATCCCAATTACGGTAACAAACAAGTCGGGGT
                    AGTCCGTCCAGTTAAGACTACGGTCTTACGAGACGCCAAC
> myname <- paste(">", month.name, sep="") # Creates sample names for the sequences.
> writeLines(as.vector(t(cbind(myname, myseq))), "myseq.fasta") # Writes the sequences in FASTA format to a file.
```

Figura 29. Texto gerado com as combinações de sequência de DNA.

>January	>July
GCTACGGTCAGCAGGCGGTGTTTTGAGGGGAGAGGGCATG	TTTGAGCCATTAACTGCATACCTTGTCTTTTCCGGCTGAT
>February	>August
TGCCGTTACCGCCGTTTGGCTCCTAACCGGGCCACCATCA	GTATATTTCCTGGATTTCATTCACGTAGCGTCCAGACATC
>March	>September
CCGCCCAATCGAGGCCTTGTCATCGAGTGCGTCGTATCTA	AAAGTGATACAGGAGAGGTGGTCGGAGGAAGATTTGCCGC
>April	>October
CGTCTTCATGATAAACATGGAGCCCAGTGTGTAATATCCT	AGTTCTTAGAGTACAAACGAGGGCACTGCAGATAATATAT
>May	>November
TATTCGGTCACTACCGTGTCGTAAAAGGTCGCCGCCCTTA	GGAAACCTTCCCCTTGCTAGTCAAATCCCGGGATCATTGC
>June	>December
TTACTAGCGCCGGAGGCCGTTTCCAGAATAGAGTCCCAAA	TAAAAGAAGTGCGTGGCTTGAAGTAATTCCCTGAATCTCA

Figura 30. Arquivo gerado em formato FASTA denominado myseq.fasta.

6.3 Verificação de Proveniência coletada sobre os experimentos no SisGExp

Na Figura 31 é exibida a tela do SisGExp com os resultados relacionados aos experimentos realizados na seção anterior. Nessa tela são exibidas as seguintes informações: (1) nome do experimento; (2) responsável pelo experimento; (3) o *script* processado; (4) Status da Execução (completo, erro, parcial); (5) tipo de erro (se houver erro); (6) data da execução.

Figura 31. Resultados dos testes com as três categorias de s*cripts* R. São informados os dados de proveniência e possíveis erros coletados durante a execução do *Script* R.

O Quadro 6 apresenta exemplos de consultas que recuperam os descritores de proveniência (prospectiva e retrospectiva) coletados pela arquitetura RFlow através do aplicativo SisGExp durante a modelagem e execução dos experimentos.

Nas subseções abaixo são exibidas consultas SQL referentes às proveniências prospectiva e retrospectiva. Essas consultas são realizadas diretamente nos esquemas “expdados” e “public”. As consultas SQL são baseadas no Quadro 6.

Quadro 6. Questões sobre proveniências prospectiva e retrospectiva que podem ser respondidas pelo pesquisador com auxílio da RFlow

Proveniência prospectiva	Proveniência retrospectiva
Fase de planejamento e acompanhamento do experimento. Nessa fase o pesquisador interage com o SisGExp e pode desejar responder às questões:	Depois de invocar o ExecScript através do SisGExp, o pesquisador é capaz de responder às seguintes questões:
1) Quem é o responsável pela modelagem do experimento? 2) Quais são os fatores, variáveis resposta e delineamento envolvidos no experimento? 3) Qual a data inicial e data final da instalação do experimento? 4) Qual o objetivo do experimento? 5) Quais são as atividades do experimento e seus status? 6) Quais publicações estão associadas ao experimento?	1) Quais são os resultados produzidos pelo experimento e análises estatísticas do experimento? 2) Quem é o responsável pela execução do *workflow* estatístico (*script*)? 3) Qual a data da execução do *workflow*? 4) Em qual servidor foi executado o *workflow*? 5) Qual o banco de dados utilizado no experimento? 6) Qual status da execução (é execução parcial, houve erro, foi completa)?

6.3.1 Exemplo de Consulta sobre proveniência prospectiva

Suponha que um pesquisador deseja consultar os descritores sobre um dado experimento. Em linguagem natural ele deseja saber:

"Quais experimentos agrícolas foram instalados durante o ano de 2014 e que tenham publicações associadas. Deseja-se obter o nome do responsável pelo experimento, o nome do experimento, suas datas de início e término da instalação, bem como seus objetivos e publicações".

Esta indagação pode ser convertida em linguagem SQL e aplicada ao esquema *expdados* que armazena dados de proveniência prospectiva. O resultado é exibido na Figura 32.

Em SQL temos:

```
SELECT usu.nome, exp.nome, exp.dt_inicial_instalacao, exp.dt_final_instalacao, trab.titulo
FROM expdados.experimento exp, expdados.usuario usu, expdados.trabalho_cientifico trab
WHERE (exp.id_responsavel=usu.id)
AND (exp.id=trab.id_experimento)
AND exp.dt_inicial_instalacao >= '2014-01-01 00:00:00'
AND exp.dt_inicial_instalacao <= '2014-12-31 00:00:00';
```

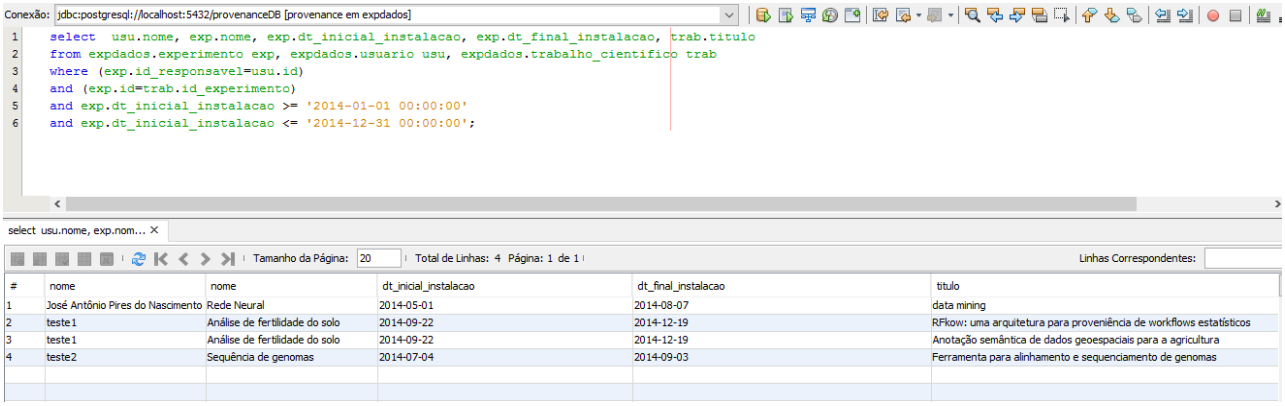

Figura 32. Consulta em SQL – Proveniência Prospectiva.

6.3.2 Exemplo de Consulta sobre proveniência retrospectiva

Suponha que o pesquisador deseja obter dados sobre a execução de um experimento computacional. Em linguagem natural ele necessita saber:

"Quais *workflows* foram executados no ano de 2015 e quais são os status da execução. Deseja-se obter as seguintes informações: nome do *workflow* concreto, nome do usuário que executou o *workflow*, anotação, período, status e a mensagem de erro se houver".

Essa consulta é realizada no esquema *public*. O resultado é exibido na Figura 33. Vale ressaltar que o meta-*workflow* ExecScript é repetido em todas as instâncias. Isso é devido o meta-*workflow* ser o responsável pelo encapsulamento de todos os *scripts* R processados. Ele age como um *workflow* estatístico genérico.

Em SQL temos:

```
SELECT w."name", exe."user", exe.annotation, exe."type", exe.start_time, exe.end_time, e.message
FROM public.workflow w, public.workflow_exec exe FULL JOIN public.error e ON exe.id=e.exec_id
WHERE w.id=exe.wf_id
AND exe.start_time >= '2015-01-01 00:00:00'
AND exe.end_time <= '2015-12-31 00:00:00';
```

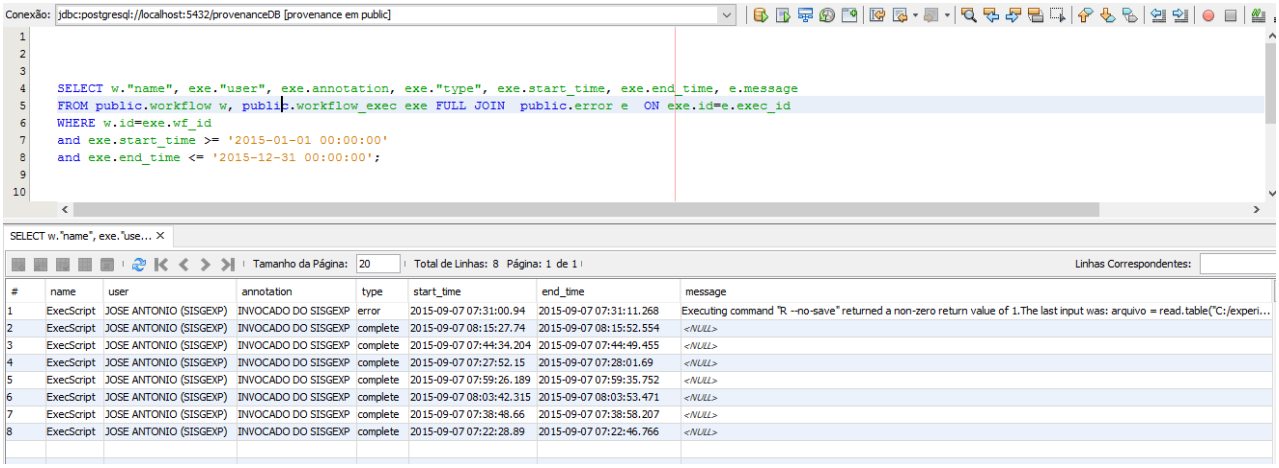

Figura 33. Consulta em SQL – Proveniência Retrospectiva.

6.3.3 Exemplo de Consulta sobre proveniência prospectiva e retrospectiva

Suponha que um pesquisador deseja consultar simultaneamente descritores de proveniência prospectiva e retrospectiva relacionados com um único experimento. Em linguagem natural ele necessita saber:

"Quais experimentos foram executados no ano de 2015 e o status de execução de cada instância. Deseja-se as seguintes informações: nome do responsável pelo experimento, nome do experimento, data de início e data de término da execução, status e a mensagem de erro (se houver erro)".

Essa consulta é realizada nos dois esquemas *expdados* e *public* e utiliza os dois tipos de proveniência coletados pela arquitetura RFlow através do SisGExp. O resultado é exibido na Figura 34.

Em SQL temos:

```
SELECT usu.nome, exp.nome, exe."type", exe.start_time, exe.end_time, e.message
FROM public.workflow w, expdados.wf_script wf, expdados.script sc, expdados.experimento exp, expdados.usuario usu, public.workflow_exec exe full join public.error e on (exe.id=e.exec_id)
WHERE (exe.id=wf.id_workflow_exec)
AND (wf.id_script=sc.id)
AND (sc.id_experimento=exp.id)
AND (exp.id_responsavel=usu.id)
AND exe.start_time >= '2015-01-01 00:00:00'
AND exe.end_time <= '2015-12-31 00:00:00';
```

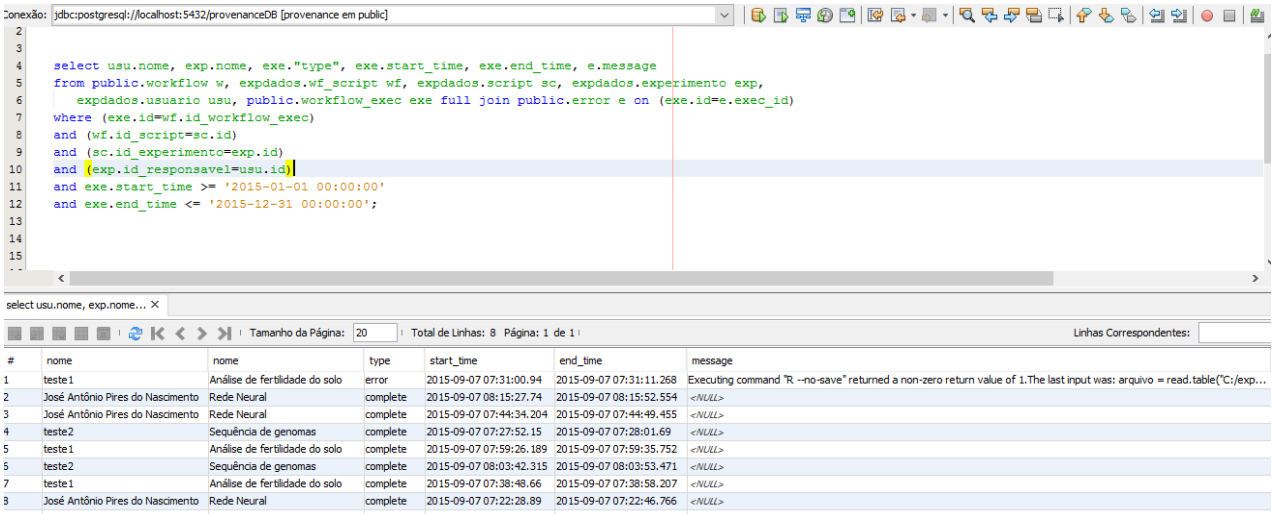

Figura 34. Consulta em SQL sobre os dois tipos de Proveniência: Prospectiva e Retrospectiva.

6.4 Discussão dos Resultados

Foi apresentado através de testes que a arquitetura RFlow é capaz de atender aos requisitos de coleta de proveniência prospectiva e retrospectiva de experimentos científicos baseados em *scripts* R legados encapsulados por meta-*workflows*.

Outros aspectos relevantes citados na hipótese dessa dissertação são: o acesso remoto, coleta de resultados *online*, tempo e esforço reduzidos. Por simplificação, a verificação da hipótese foi dividida em três fragmentos:

a) recuperação e reprodutibilidade dos experimentos e seus resultados estatísticos

A reprodutibilidade do experimento baseado em *scripts* R legados é possível devido a integração do SisGExp com o SGWfC Kepler. O Kepler é o responsável pelo gerenciamento e execução do experimento em conjunto com o sistema R e o meta-*workflow* ExecScript. Durante a execução do ExecScript, além de gerar os resultados científicos *online*, como apresentado na seção anterior, é registrada a proveniência retrospectiva (resultados estatísticos) e vinculada com o experimento cadastrado anteriormente (proveniência prospectiva) no banco de dados provenanceDB.

b) acessibilidade da solução em qualquer Unidade da Embrapa

O SisGExp é um aplicativo Web, portanto está hospedado em um servidor de aplicação Web, no caso o servidor GlassFish, que é capaz de executar de modo ininterrupto em regime de 24x7 e estará disponível para acessos a partir de um navegador de páginas que o usuário utilize. O aplicativo, uma vez instalado em um servidor de domínio público (internet), pode ser acessado de qualquer lugar que possua conectividade. Para isso, é preciso apenas

digitar em um navegador de páginas a *url* de onde está localizado o aplicativo.

c) tempo e esforço reduzidos

A arquitetura RFlow garante que as informações requeridas pelos pesquisadores podem ser disponibilizadas de forma rápida e com baixo esforço devido os resultados estatísticos serem gerados *online*. A funcionalidade do SisGExp de executar os *scripts* R legados através da "invocação" do Kepler é o recurso que proporciona baixo esforço cognitivo para ter acesso aos resultados estatísticos. Uma vez que não é necessário a redigitação de dados. Além disso, o pesquisador é capaz de baixar os resultados para sua máquina local, ou seja, os resultados são gerados automaticamente quando o pesquisador seleciona o *script* R para executar.

6.5 Considerações Adicionais

O aplicativo SisGExp será possivelmente o principal responsável por gerenciar os descritores coletados sobre os experimentos agrícolas realizados pela Embrapa. Isso possibilitará a redução total ou parcial para a falta de padrão de informações referentes à coleta de dados agrícolas, que muito prejudicam a atividade do profissional de Estatística na elaboração dos trabalhos de cunho estatístico. Além da padronização dos dados, outro aspecto positivo que o aplicativo apresenta é a organização das informações de pesquisa em uma base de dados única, que poderá reduzir consideravelmente a redundância de experimentos e dados.

Segundo o comunicado técnico da Embrapa sobre Banco de Dados de Experimentos Agrícolas, Análise e Projeto (SILVA *et. al*, 2001), há uma grande preocupação com a falta de padrão na coleta e tratamento dos dados agrícolas. Através de uma base de dados de experimentos agrícolas bem organizada, será possível utilizar e desenvolver novas análises matemáticas e estatísticas. Com isso, servir de recomendação agrícola para várias regiões do Brasil e, assim, tornar-se uma base a ser utilizada por sistemas especialistas e sistemas de modelagem e simulação.

Os seguintes serviços podem ser oferecidos pelo sistema SisGExp no ambiente da Embrapa:

- Fornecimento de um serviço informativo a órgãos de financiamento de pesquisa - estado da arte e prioridades de investimento;
- Banco de dados agrícola confiável e atualizado por pesquisadores e entidades afins que será utilizado por eles para análise de tendências gerais e observações na pesquisa;
- Redução de redundância de experimentos agrícolas afins;
- Maior integração de equipes temáticas dispersas nas diversas Unidades da Empresa.

7 CONCLUSÕES

Esse trabalho apresentou a arquitetura RFlow, um conjunto de ferramentas integradas, que possibilita ao pesquisador cadastrar seus experimentos e publicações científicas, compartilhá-los com a comunidade científica, gerar resultados estatísticos *online*, e ainda permite coletar diferentes tipos de proveniência sobre os experimentos agropecuários através do uso do SGWfC Kepler e o meta-*workflow* ExecScript.

A arquitetura oferece abstração na definição de experimentos e novas facilidades na possibilidade de reproduzir e rastrear esses experimentos, graças aos mecanismos de coleta de proveniência presentes no SGWfC Kepler e no aplicativo SisGExp. Acredita-se que a ferramenta é robusta e capaz de atender experimentos de outros domínios da ciência além dos experimentos agropecuários. No entanto, se ressalta que mais testes exaustivos devem ser realizados.

Um dos grandes diferenciais desse trabalho é a vinculação das publicações científicas como teses, dissertações, artigos, entre outros com o conjunto de dados informados nas fases de planejamento e acompanhamento do experimento. Isso permitirá a validação dos resultados estatísticos constantes nas publicações, bem como a reprodução do experimento por outros pesquisadores, agências de fomento e editores de revistas localizados em toda parte do globo terrestre que tenham acesso à internet.

A solução apresentada permite que a execução do *Script* R legado bem como a coleta de proveniências sejam realizadas sem haver necessidade de modificações no *workflow* ou conhecimento da linguagem R. Isso vai diminuir o tempo, custo e esforço do pesquisador para simular um ambiente de experimentação.

7.1 Contribuições

As principais contribuições desse trabalho são: a definição de uma arquitetura denominada RFlow e sua implementação, o aplicativo java web SisGExp (está na versão Beta), um meta-*workflow* genérico ExecScript para o SGWfC Kepler capaz de encapsular *scripts* R legados e um *schema* de dados OPM-compatível, cujo propósito é armazenar os diferentes tipos de proveniência.

7.2 Limitações

Todo trabalho científico possui limitações. Nesse caso, o escopo desse trabalho está baseado em análises estatísticas feitas na linguagem R, portanto quando se menciona o termo *script* ou *workflow* estatístico ou análises estatísticas ou resultados estatísticos são todos baseados apenas na linguagem R. Mais investigações futuras são necessárias para as demais linguagens estatísticas.

O foco principal da arquitetura RFlow é trabalhar com experimentos agrícolas, embora em algumas ocasiões são citados experimentos agropecuários na dissertação, porém isso não compromete o estudo, uma vez que a funcionalidade de leitura e gravação de *scripts* R e

arquivos de dados do aplicativo SisGExp está baseada na formatação necessária aos dois tipos de experimentos.

Os erros referentes às falhas de execução do *script* R legado são registrados no banco de dados provenanceDB e disponibilizados para o usuário consultar e avaliar os fatores limitantes. Caso o *script* R possua alguma interação com o usuário, por exemplo, entrada de dados via teclado, será emitido um erro na execução desse *script*, pois a arquitetura processa somente *scripts* do tipo não interativo.

A arquitetura não calcula ou realiza análises estatísticas, ela requer que tais artefatos já estejam prontos no *script* R legado. No entanto, isso não impede que o pesquisador altere os parâmetros da planilha de dados ou baixe novos *scripts* R e altere alguns procedimentos, e após isso, reexecute o experimento.

A intervenção do usuário na execução de *scripts* R se limita à instalação e carregamento de pacotes no sistema R, definição da estrutura de pastas de trabalho (execução local) e a formatação adequada da planilha de dados utilizada pelo *script*.

O ator "Provenance Recorder" versão 4.1 não trabalha com esquemas de banco de dados no SGBD PostgreSQL, ou seja, somente aceita o esquema padrão *public*. O padrão de arquivos do PR é o OPM.

7.3 Trabalhos Futuros

A arquitetura RFlow, especificamente o aplicativo SisGExp está na versão beta, portanto são necessários mais testes para consolidar a ferramenta. Como futuros desenvolvimentos têm-se interesse em ampliar o leque de tipos de SGWfC a serem utilizados pela arquitetura RFlow. Para isso, deverá ser customizado o serviço de configuração da arquitetura. Além disso, há interesse nas demandas abaixo:

- Criar uma interface no SisGExp com a linguagem R para possibilitar que o pesquisador faça algumas análises estatísticas diretamente no SisGExp. Isso permitirá que os pesquisadores façam análises estatísticas sem conhecer o R;
- Gerar mais relatórios no SisGExp ou permitir através de novas funcionalidades no sistema que o próprio pesquisador construa novos relatórios;
- Avaliar a integração da arquitetura com outros softwares livres de diferentes domínios (biologia, química, geoprocessamento, bioinformática).

8 REFERÊNCIAS BIBLIOGRÁFICAS

AALST, W. V. D.; HOFSTEDE, A.; KIEPUSZEWSKI, B.; BARROS, A. "Workflow patterns", ***Distributed and Parallel Databases*****,** v. 14, n. 1, p. 5-51. 2003.

ABOUELHODA, M.; ISSA, S. A.; GHANEM, M. Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support. **BMC Bioinformatics**, *13*, 77. 2012.

ALTINTAS, I. et al. "Provenance Collection Support in the Kepler Scientific Workflow System", **IPAW2006**, 118-132, 2006.

ALTINTAS, I.; BERKLEY, C.; JAEGER, E.; JONES, M.; LUDASCHER, B.; MOCK, S. "Kepler: an extensible system for design and execution of scientific workflows". **Scientific and Statistical Database Management**, p. 423-424, Greece.2004.

ANDERSON, C. The end of theory: the data deluge makes the scientific method obsolete. **Wired Magazine**, 23 jun. 2008. Disponível em: <http://www.wired.com/science/discoveries/magazine/16-07/pb_theory>. Acesso em 25 mar. 2013.

APACHE.Commons.2014.[S.l.]. Disponível em < https://commons.apache.org/proper/commons-io/>. Acessado em: 04 de Nov. de 2014.

ATKINSON, M.; BRITTON, D.; COVENEY, P.; DE ROURE, D.; GARNETT, N.; GEDDES, N., GURNEY, R.; HAINES, K.; HUGHES, L.; INGRAM, D.; JEFFREYS, P.; LYON, L.; OSBORNE, I.; PERROTT, R.; PROCTER, R.; RUSBRIDGE, C.; TREFETHEN, A. A; WATSON, P. Century-of-Information Research (CIR): a strategy for research and innovation in the Century of Information. **Prometheus**, v. 27, n. 1, p. 27-45, 2009.

BANZATTO, D.A.; KRONKA, S. N. EXPERIMENTAÇÃO AGRÍCOLA. Jaboticabal. SP. **FUNEP**. 1992.

BARROS, A. J. P.; LEHFELD, N. A. S. Projeto de Pesquisa: Propostas Metodológicas. 8 a. ed. Petrópolis. **Vozes**, 95 p. 1999.

BAUMER, B.; CETINKAYA-RUNDEL, M.; BRAY, A.; LOI, L.; HORTON, N. J. R. markdown: Integrating a reproducible analysis tool into introductory statistics. **ArXiv e-prints**, February 2014.

BIOINFORMATICS, manuals.[S.l.].2015, Disponível em <http://manuals.bioinformatics.ucr.edu/home/ht-seq>. Acessado em: 13 Mar. 2015.

BUNEMAN, P.; KHANNA, S. E.; CHIEW, W. Why and Where: a Characterization of Data Provenance. **ICDT'01: 8th International Conference on Database Theory, LNCS**, v.1973, p.316–330, 2001.

CALLAHAN, S. P.; FREIRE, J.; SANTOS, E.; SCHEIDEGGER, C. E.; SILVA, C. T.; VO, H. T. "VisTrails: visualization meets data management". **SIGMOD**, p. 745-747, Chicago, Illinois, USA.2006.

CASADEVALL, A.; FANG, F. C. Infect Immun. Reproducible Science. doi: 10.1128/IAI.00908-10 **PMCID: PMC2981311**.2010

CESARIO, E.; LACKOVIC, M.; TALIA, D.; TRUNFIO, P. "Service-oriented data analysis in distributed computing systems," in High Performance Computing: From Grids and Clouds to Exascale, Eds., pp. 225–245, **IOS Press, Lansdale**, Pa, USA. 2011.

CHAMBERS, J. R. Software Data Analysis Programming with R Software. **Springer**. 1st edition, 2008.

COHEN, S.; BOULAKIA, S. E.; DAVIDSON, S. Towards a Model of Provenance and User Views in Scientific Workflows, Data Integration in the Life Sciences, **LNCS 4075, Springer**, p.264–279, 2006.

CRAWLEY, M. J. Statistical Computing to Data Analysis using S-plus. **Wiley**. 1st edition, 2002.

CRUZ, S. M. S.; CAMPOS, M. L M.; MATTOSO, M. L. Q. "Towards a Taxonomy of Provenance in Scientific Workflow Management Systems". **Services**.pp. 259 – 266. 2009.

CRUZ, S. M. S. "Uma Estratégia De Apoio À Gerência De Dados De Proveniência Em Experimentos Científicos". Tese de Doutorado, **COPPE/UFRJ**. 2011.

DEELMAN, E.; GANNON, D.; SHIELDS, M.; TAYLOR, I. Workflows and e-Science: An overview of workflow system features and capabilities. **Future Generation Computer Systems**. v. 25, n. 5, p. 528-540. 2009.

DEELMAN, E.; SINGH, G.; SU, M. H. et al., "Pegasus: a framework for mapping complex scientific workflows onto distributed systems" **Scientific Programming**, vol. 13, no. 3, pp. 219–237, 2005.

ELLKVIST, T.; KOOP, D.; ANDERSON, E. W.; FREIRE, J.; SILVA, C. "Using Provenance to Support Real-Time Collaborative Design of Workflows", Provenance and Annotation of Data and Processes: 2nd International Provenance and Annotation Workshop, Salt Lake City, UT, USA, L/CS, **Springer-Verlag**, p. 266-279. 2008.

EMBRAPA. (2015).**Empresa Brasileira de Pesquisa Agropecuária**. Disponível em: <https://www.embrapa.br/quem-somos>. Acesso em: 7 Mar. 2015.

EMBRAPA INFORMÁTICA AGROPECUÁRIA.(2015a).**Embrapa Informática Agropecuária**. Disponível em: <https://www.embrapa.br/informatica-agropecuaria/missao-visao-valores>. Acesso em: 7 Mar. 2015.

EMBRAPA INFORMÁTICA AGROPECUÁRIA. (2015b).**Embrapa Informática**

Agropecuária. Disponível em: <https://www.embrapa.br/informatica-agropecuaria/infraestrutura/laboratorio-multiusuario-de-bioinformatica>. Acesso em: 7 Mar. 2015.

FREIRE, J.; KOOP, D.; SANTOS, E.; SILVA, C. T. "Provenance for Computational Tasks: A Survey", **Computing in Science and Engineering**, v.10, n. 3, p. 11-21. 2008.

FREIRE, J.; BONNET, P.; SHASHA, D. Computational reproducibility: state-of-the-art, challenges, and database research opportunities "http://dl.acm.org/img/shopping-art16.gif" ;**New York University,** Poly, Brooklyn, NY, USA; IT University of Copenhagen, Copenhagen, Denmark. 2012.

GEKKOQUANT. Gekkoquant. Disponível em: <http://gekkoquant.com/2012/05/26/neural-networks-with-r-simple-example/,Neural Networks with R – A Simple Example>. Acesso em: 17 Jun. 2015.

GIARDINE, B.; RIEMER, C.; HARDISON, R. C.; BURHANS, R.; ELNITSKI, L.; SHAH, P.; ZHANG, Y.; BLANKENBERG, D.; ALBERT, I.; TAYLOR, J.; MILLER, W.; KENT, W. J.; NEKRUTENKO, A. "Galaxy: a platform for interactive large-scale genome analysis." **Genome Research**. 2005.

GOBLE, C. A.; BHAGAT, J.; ALEKSEJEVS, S.; CRUICKSHANK, D.; MICHAELIDES, D.; NEWMAN, D.; BORKUM, M.; BECHHOFER, S.; ROOS, M. myExperiment: a repository and social network for the sharing of bioinformatics workflows, NucleicAcids Research, v. 38, n. **Web Server Issue**. p. 677-682. 2010.

GOMES, F. P. **Curso de estatística experimental**. 13. ed., Piracicaba: **Nobel**, 1996.

GONZÁLEZ-BELTRÁN, A.; LI, P.; ZHAO, J.; AVILA-GARCIA, M. S.; ROOS, M.; THOMPSON, M. et al. From Peer-Reviewed to Peer-Reproduced in Scholarly Publishing: The Complementary Roles of Data Models and Workflows in Bioinformatics. **PLoS ONE** 10(7): e0127612. doi:10.1371/journal.pone. 2015.

GRAY, J. Jim Gray on escience: a transformed scientific method. In: HEY, T.; TANSLEY, S.; TOLLE, K. (Ed.). The fourth paradigm: data-intensive scientific discovery. Washington: **Microsoft Research**, 2009.

GUERRA, M. J.; DONAIRE, D. **Estatística intuitiva**. 5 ed. São Paulo: LTC, 1991.

HEY, T.; TANSLEY, S.; TOLLE, K., **The Fourth Paradigm: Data-Intensive Scientific Discovery**. 1. ed. Redmond, Microsoft Research, 2009.

HIGGINS, D. Using R in Kepler, **Berkeley University**, <ptolemy.eecs.berkeley.edu/conferences/05/presentations/higginsRSystem.pdf>, 2007.

HINKELMANN, K.; KEMPTHORNE, O. Design and analysis of experiments. New York: **J. Wiley**,. 631 p. 1994.

HOFFMANN, R; VIEIRA, S. Estatística experimental. São Paulo: **Atlas**, 1989.

HULL, D.; WOLSTENCROFT, K.; STEVENS, R.; GOBLE, C.; POCOCK, M. R.; LI, P.; KEPLER, 2013. Disponível em: <https://code.kepler-project.org/code/kepler/trunk/modules/provenance/docs/provenance.pdf>. Acesso em: 23 Fev. 2013

KIRCHKAMP, O. "Workflow of statistical data analysis". Disponível em: <http://www.kirchkamp.de/oekonometrie/pdf/wf-screen2.pdf>. Acesso em: 05 Out. 2014

KUMAR, A.; WAINER, J. "Meta-workflows as a control and coordination mechanism for exception handling in workflow systems". **Decision Support Systems**. v. 40 pp. 89-105.2005.

LAKATOS, E. M.; MARCONI, M. A. Metodologia Cientifica. 2a . ed. São Paulo: **Editora Atlas.** 242 p. 1991.

LERNER, B.; BOOSE, E. RDataTracker: Collecting Provenance in an Interactive Scripting Environment. In 6th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2014), Cologne, **USENIX Association**. 2014.

LI, Q.; BROWN, J. B.; HUANG, H.; BICKEL, P. J. MEASURING REPRODUCIBILITY OF HIGH-THROUGHPUT EXPERIMENTS, **The Annals of Applied Statistics**, Vol. 5, No. 3, 1752–1779. 2011.

LITTAUER, R.; RAM, K.; LUDÄSCHER, B.; MICHENER, W.; KOSKELA, R.Trends in Use of Scientific Workflows: Insights from a Public Repository and Recommendations for Best Practice.**Int J Digit Curation.**7(2):92-100. 2012.

LOANNIDIS, J. P. A. PLoS Med.2005:e124. Why most published research findings are false.**Epub**.2005.

LUDÄSCHER, B. et al. "Scientific workflow management and the Kepler system: Research Articles". **Concurrency and Computation: Practice & Experience,** v. 18, n. 10, p. 1039-1065, 2006.

MAIR, P.; DE LEEUW, J. "A general framework for multivariate analysis with optimal scaling: The R package aspect". **Journal of Statistical Software**, 32(9), pp. 1-12, 2010.

MARCONI, M.; LAKATOS, E. M. *Fundamentos de metodologia científica*. 7.ed. São Paulo: **Atlas**, 2010.

MARINHO, A.; MURTA, L.; WERNER, C.; *et al.*., "Integrating Provenance Data from Distributed Workflow Systems with ProvManager". In: **Provenance and Annotation of Data and Processes**, v. 6378, Lecture Notes in Computer Science. Springer, pp. 286-288, 2010.

MATES, P.; SANTOS, E.; FREIRE, J.; SILVA, C. T. CrowdLabs: Social Analysis and Visualization for the Sciences. In: **23rd Scientific and Statistical Database Management Conference23rd Scientific and Statistical Database ManagementConference**, Portland,

Oregon, USA, 2011.

MATTOSO, M.; CRUZ, S. M. S. Gerência de workflows científicos: oportunidades de pesquisa em bancos de dados. In: **Proceedings of the 23rd Brazilian symposium on Databases**, pp. 313-314, Campinas, Sao Paulo, Out. 2008

MATTOSO, M.; WERNER, C.; TRAVASSOS, G. H.; *et al.* Gerenciando Experimentos Científicos em Larga Escala. In: **Anais do XIII Congresso da Sociedade Brasileira de Computação**, pp. 121-135, Belém, Jul. 2008.

MATTOSO, M.; et al. "Desafios no apoio à composição de experimentos científicos em larga escala". In: **Seminário Integrado de Software e Hardware (XXXVI SEMISH)**, pp. 307-321, 2009.

MCPHILLIPS, T. M.; SONG, T.; KOLISNIK, T.; AULENBACH, S.; et al. Yesworkflow: A user-oriented, language-independent tool for recovering workflow information from scripts. **CoRR, abs**/1502.02403, 2015.

MOREAU, L.; FREIRE, J.; MYERS, J.; FUTRELLE, J.; PAULSON, P. *The Open Provenance Model*, **Technical report, Electronics and Computer Science,** University of Southampton. 2007.

MOREAU, L.; MISSIER, P.; BELHAJJAME, K.; CRESSWELL, S.; GOLDEN, R.; GROTH, P.; MILES, S.; SAHOO, S. (2011). **The PROV Data Model and Abstract Syntax Notation**. Disponível em: http://www.w3.org/TR/prov-dm/. Acesso em: 17 Mar. 2014.

MYGRID.2008. Disponível em: <http://www.mygrid.org.uk/>. Acesso em: 01 jul. 2015.

MURTA, L.; BRAGANHOLO, V.; CHIRIGATI, F.; KOOP, D.; FREIRE, J. noWorkflow: Capturing and Analyzing Provenance of Scripts. **5th International Provenance and Annotation Workshop, IPAW.** LNCS. Vol. 8628, p 71-83. 2014.

NASCIMENTO, J. A. P.; CRUZ, S. M. S. RFlow: Uma Abordagem de Reutilização de Workflows Estatísticos Legados. In: Maceió - Alagoas. **XXXIII Congresso da Sociedade Brasileira de Computação, VII e-Science workshop**, 2013.

NASCIMENTO, J. A. P.; CRUZ, S. M. S. RFlow: uma arquitetura para proveniência de *workflows* estatísticos. In: Curitiba - Paraná. **X Congresso Brasileiro de Agroinformática, SBIAGRO**.2015.

NAGAVARAM, A.; AGRAWAL, G.; FREITAS, M.; MEHTA, G.; MAYANI, R.; DEELMAN, E.“A cloud-based dynamic workflow for mass spectrometry data analysis,” in **Proceedings of the 7th IEEE International Conference on e-Science (e-Science '11**), December 2011.

NOBELPRIZE.2013. Disponível em <http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/popular-chemistryprize2013.pdf>. Acesso em: 21 Jun. 2014.

NOGUEIRA, M. C. S. Estatística experimental aplicada à experimentação agrícola. Piracicaba: **USP-ESALQ**, 250 p. 1997.

OINN, T.; LI, P.; KELL, D. B.; GOBLE, C.; GODERIS, A.; GREENWOOD, M.; HULL, D.; STEVENS, R.;TURI, D.; ZHAO, J. Taverna/myGrid: Aligning a Workflow System with the Life Sciences Community, Workflows for e-Science, **Springer**, p. 300-319, 2007.

OINN, T. "Taverna: a tool for building and running workflows of services", **Nucleic Acids Research**, v. 34, n. 2, p. 729-732. 2006.

PENG, R. D. Reproductible Research in Computer Science, **Science,** Vol. 334 no. 6060 p. 1226-1227, 2011.

POPPER, K. R.The logic of scientific discovery. Hutchinson, London, **United Kingdom**. 1959.

POSTGRESQL, (2009), PostgreSQL, Disponível em < http://www.postgresql.org>.Acessado em: 03 Jan. 2014.

PRIMEFACES, (2009), Disponível em <http://primefaces.org/downloads>. Acessado em: 25 Out. 2014.

QIN, Z.; XING, J.; ZHENG, X. Software architecture. **Springer**. 1st edition.2008.

RANABAHU, A.; ANDERSON, P.; SHETH, A. P. "The Cloud Agnostic e-Science Analysis Platform". **IEEE Internet Computing** v. 15.pp. 85-89. 2011.

R DEVELOPMENT CORE TEAM. **The R project for statistical computing**. Vienna, 2012. Disponível em: < http://www.R-project.org>. Acesso em: 17 Mar. 2013.

RUNNALLS, A. "CXXR: an extensible R interpreter In: **Wiley Interdisciplinary Reviews: Computational Statistics**. DOI: 10.1002/wics.1251, 2013.

RUSSELL, N.; HOFSTEDE, A.; AALST, W. V. D; MULYAR, N. "Workflow control-flow patterns: A revised view", **BPM Center Report BPM-06-22, BPMcenter.org**, p. 06–22. 2006.

SILLES, C. A.; RUNNALLS, A. "Provenance-Awareness in R". **LNCS**, vol. 6378, p. 64-72, 2010.

SILVA, C. E. P. Captura de Dados de Proveniência de Workflows Científicos em Nuvens Computacionais / Carlos Eduardo Paulino Silva. – Rio de Janeiro: **UFRJ/COPPE**, 2011.

SILVA, F. C. D; ADACHI, D. T.; NARCISO, M. G; JÚNIOR, V. B. **Banco de Dados de Experimentos Agrícolas: Análise e Projeto**. Campinas: Embrapa Informática Agropecuária, (**Embrapa Informática Agropecuária. Comunicado Técnico, 6**). 2001.

TALIA, D.; TRUNFIO, P.; VERTA, O. "Weka4WS: a WSRF-enabled Weka toolkit for

distributed data mining on Grids," in **Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases**, pp. 309–320, Porto, Portugal, 2005.

TALIA, D. "Workflow Systems for Science: Concepts and Tools", I**SRN Software Engineering**, vol. 2013, Article ID 404525, 15 pages, doi:10.1155/2013/404525. 2013.

TAYLOR, I.; SHIELDS, M.; WANG, I.; RANA, O. "Triana, applications within Grid computing and peer to peer environments", **Journal of Grid Computing**, vol. 1, pp. 199–217, 2004.

TAYLOR, I.; DEELMAN, E.; GANNON, D.; et al. Workflows for e-Science: Scientific Workflows for Grids. 1 ed. London, **Springer-Verlag**, 2007.

TRAVASSOS, G. H.; BARROS, M. O. "Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering". In: **Proceedings of the WSESE03**, pp. 189-200, Roma, Ago. 2003.

TUOT, C. J.; SINTEK, M.; DENGEL, A. R. IVIP – A Scientific Workflow System to Support Experts in Spatial Planning of Crop Production. **Scientific and Statistical Database Management. LNCS**, vol. 5069, p 586-591. 2008.

UNICAMP. Campinas, SP, 2015, Disponibilizado em <http://www.unicamp.br/iq/cces/public/index.php>. Acessado em: 15 Jan. 2015

VAZ, G. J. e-Science na Embrapa / José Glauber Vaz. - Campinas: **Embrapa Informática Agropecuária**, 2011.

VISTRAILS.**VisTrails Documentation**., 2013. Disponível em: <http://www.vistrails.org/usersguide/v2.0/html/VisTrails.pdf>. Acesso em: 16 set. 2014

VÖCKLER, J. S.; JUVE, G.; DEELMAN, E.; RYNGE, M.; BERRIMAN, B. "Experiences using cloud computing for a scientific workflow application," in **Proceedings of the 2nd International Workshop on Scientific Cloud Computing (ScienceCloud '11)**, pp. 15–24,.View at Publisher·View at Google Scholar·View at Scopus. June 2011.

WASHINGTON. 2015.University of Washington Escience Institute. Washington, 2015. Disponibilizado em <http://escience.washington.edu/>. Acessado em: 15Mar. 2015.

W3C. PROV-DM: The PROV Data Model. 2012. Disponvel em: <www.w3.org/TR/provdm/ >.Acessado em: 13 Maio de 2014.

WILSON J. E. B., An Introduction to Scientific Research. 2. ed. **Dover Publications**, 1991.

ZHAO, J.; GOBLE, C.; STEVENS, R.; BECHHOFER, S. "Semantically linking and browsing provenance logs for e-science", **Semantics of a Networked World**, v. 3226, p. 158–176. 2004.

ZHAO, Z.; PASCHKE, A. A. Survey on Semantic Scientific Workflow Semantic Web Journal, **IOS press** 1-5. 2012

9 ANEXOS

ANEXO A - Categoria 1: Treinar uma rede para calcular a raiz quadrada de números entrados aleatoriamente – utiliza recursos de redes neurais

```
# http://gekkoquant.com/2012/05/26/neural-networks-with-r-simple-example/

# install.packages('neuralnet')

# Exemplo_mlp_square

library("neuralnet")

 #Going to create a neural network to perform square rooting

#Type ?neuralnet for more information on the neuralnet library

 #Generate 50 random numbers uniformly distributed between 0 and 100

#And store them as a dataframe

traininginput <-  as.data.frame(runif(50, min=0, max=100))

trainingoutput <- sqrt(traininginput)

 #Column bind the data into one variable

trainingdata <- cbind(traininginput,trainingoutput)

colnames(trainingdata) <- c("Input","Output")

 #Train the neural network

#Going to have 10 hidden layers

#Threshold is a numeric value specifying the threshold for the partial

#derivatives of the error function as stopping criteria.

net.sqrt <- neuralnet(Output~Input,trainingdata, hidden=10, threshold=0.01)

print(net.sqrt)

 #Plot the neural network

plot(net.sqrt)
```

```
 #Test the neural network on some training data
testdata <- as.data.frame((1:10)^2) #Generate some squared numbers
net.results <- compute(net.sqrt, testdata) #Run them through the neural network
 #Lets see what properties net.sqrt has
ls(net.results)
 #Lets see the results
print(net.results$net.result)
 #Lets display a better version of the results
cleanoutput <- cbind(testdata,sqrt(testdata),
                                          as.data.frame(net.results$net.result))
colnames(cleanoutput) <- c("Input","Expected Output","Neural Net Output")
print(cleanoutput)
```

ANEXO B - Categoria 2: Agrobiologia (funções estatísticas)

arquivo = read.table("dadosPesqSolo.txt", h=T) # importando o arquivo do excel

arquivo # lendo o arquivo no R

dim(arquivo) # dá a dimensão do arquivo linhas x colunas

names(arquivo) # dá o nome de cada coluna

attach(arquivo) # o objeto arquivo é incluído no caminho de procura usando o comando # attach para facilitar a digitação.

is.factor(Trata) # Trata é fator Nomear na plan excel como: como Tr1, Tr2, etc

is.factor(Mat) # Mat é fator Nomear na plan excel como: como Mat1, Mat2, etc

is.numeric(PorcN) # Porcn é variável numérica

arquivo.m <- tapply(PorcN, list(Trata,Mat), mean) # calcula a média para todas interações (médias dentro da tabela)

arquivo.m # solta as médias das interações (médias de dentro da tabela de dupla entrada)

arquivo.mt <- tapply(PorcN, Trata, mean) # calcula a média geral para Trata

arquivo.mt # solta a média geral para Trata

arquivo.mm <- tapply(PorcN, Mat, mean) # calcula a média geral para Mat

arquivo.mm # solta a média geral para Mat

par(mfrow=c(1,2)) # fazer gráfico bimdimensional

interaction.plot(Trata, Mat, PorcN) # solta gráfico bimdimensional PorcN x Trat para cada Mat

interaction.plot(Mat, Trata, PorcN) # solta gráfico bimdimensional PorcN x Mat para cada Trata

arquivo.av <- aov(PorcN ~ Trata + Mat + Trata * Mat) # faz Anava conforme DIC em fatorial

arquivo.av <- aov(PorcN ~ Trata * Mat) # Se colocar só interação o R considera automaticamente os efeitos principais

summary(arquivo.av) # solta a Anava completa

ANEXO C - Categoria 3: Bioinformática (sequenciamento e alinhamento genético)

```
#####################
## String Matching ##
#####################
myseq <- c("ATGCAGACATAGTG", "ATGAACATAGATCC", "GTACAGATCAC") # Creates a sample sequence data set.
myseq[grep("ATG", myseq)] # String searching with regular expression support.
pos1 <- regexpr("AT", myseq) # Searches 'myseq' for first match of pattern "AT".
as.numeric(pos1); attributes(pos1)$match.length # Returns position information of matches.
pos2 <- gregexpr("AT", myseq) # Searches 'myseq' for all matches of pattern "AT".
as.numeric(pos2[[1]]); attributes(pos2[[1]])$match.length # Returns position information of matches in first sequence.
gsub("^ATG", "atg", myseq) # String substitution with regular expression support.
########################
## Positional Parsing ##
########################
nchar(myseq) # Computes length of strings.
substring(myseq[1], c(1,3), c(2,5)) # Positional parsing of several fragments from one string.
substring(myseq, c(1,4,7), c(2,6,10)) # Positional parsing of many strings.
############################
## Reverse and Complement ##
############################
myseq_comp <- chartr("ATGC", "TACG", myseq) # Returns complement for given DNA sequences.
```

```
substring(myseq[1], 1:nchar(myseq[1]), 1:nchar(myseq[1])) # Vectorizes single sequence.

x <- strsplit(myseq_comp, "") # Vectorizes many sequences.

x <- lapply(x, rev) # Reverses vectors.

myseq_revcomp <- sapply(x, paste, collapse="") # Collapses vectors to strings. Final result: reverse and complement of myseq.

########################################

## Translate DNA Sequence into Protein ##

########################################

y <- c("ATGCATTGGACGTTAG") # Creates sample DNA sequence.

AAdf <- read.table(file="http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/My_R_Scripts/AA.txt", header=T,"\t") # Imports genetic code.

AAv <- AAdf[,2]; names(AAv) <- AAdf[,1] # Creates named vector with genetic code

y <- gsub("(...)", "\\1_", y) # Inserts "_" after each triplet.

y <- unlist(strsplit(y, "_")) # Splits on "_" and returns vectorized triplets.

y <- y[grep("^...$", y)] # Removes incomplete triplets.

AAv[y] # Translation into protein by name-based subsetting.

############################

## Create Random Sequences ##

############################

sapply(1:100, function(x) paste(sample(c("A","T","G","C"), 20, replace=T), collapse="")) # Creates 100 random DNA sequences with 20 residues.

sapply(1:100, function(x) paste(sample(1:40, 20, replace=T), collapse=" ")) # Creates 100 random score sets with 20 elements.

####################################################

## Sequence Printing and Writing in FASTA Format ##
```

```
#######################################################
myseq <- sapply(1:12, function(x) paste(sample(c("A","T","G","C"), 40, replace=T), collapse=""))
writeLines(myseq) # Prints the sequences to the screen, one per line.
writeLines(strtrim(myseq, 10)) # The strtrim() function allows to print only the first section of the sequences.
writeLines(strwrap(myseq, indent = 20)) # The strwrap() function provides wrapping, indenting and prefixing utilities.
myname <- paste(">", month.name, sep="") # Creates sample names for the sequences.
writeLines(as.vector(t(cbind(myname, myseq))), "myseq.fasta") # Writes the sequences in FASTA format to a file.
```

ANEXO D - Esquema EXPDADOS

```
--

-- PostgreSQL database dump
--
SET statement_timeout = 0;
SET lock_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
--
-- Name: expdados; Type: SCHEMA; Schema: -; Owner: provenance
--
CREATE SCHEMA expdados;
ALTER SCHEMA expdados OWNER TO provenance;
SET search_path = expdados, pg_catalog;
SET default_tablespace = '';
SET default_with_oids = false;
--
-- Name: analise_estatistica; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE analise_estatistica (
    id integer NOT NULL,
    descricao character varying(255),
    metodo character varying(255)
);
ALTER TABLE expdados.analise_estatistica OWNER TO postgres;
--
-- Name: analise_estatistica_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE analise_estatistica_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.analise_estatistica_id_seq OWNER TO postgres;
--
-- Name: analise_estatistica_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE analise_estatistica_id_seq OWNED BY analise_estatistica.id;
--
-- Name: atividade_experimental; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE atividade_experimental (
    id integer NOT NULL,
    data_final date,
    data_inicial date,
    nome character varying(255),
    objetivo character varying(255),
    status character varying(255),
    id_experimento integer
);
ALTER TABLE expdados.atividade_experimental OWNER TO postgres;
```

```
--
-- Name: atividade_experimental_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE atividade_experimental_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.atividade_experimental_id_seq OWNER TO postgres;
--
-- Name: atividade_experimental_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE atividade_experimental_id_seq OWNED BY atividade_experimental.id;
--
-- Name: delineamento_experimental; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE delineamento_experimental (
    id integer NOT NULL,
    nome character varying,
    aplicacao character varying
);
ALTER TABLE expdados.delineamento_experimental OWNER TO postgres;
--
-- Name: delineamento_experimental_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE delineamento_experimental_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.delineamento_experimental_id_seq OWNER TO postgres;
--
-- Name: delineamento_experimental_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner:
postgres
--
ALTER SEQUENCE delineamento_experimental_id_seq OWNED BY delineamento_experimental.id;
--
-- Name: experimento; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE experimento (
    id integer NOT NULL,
    analise_resultado character varying(255),
    cod_projeto character varying(255),
    data_coleta date,
    dt_final_instalacao date,
    dt_inicial_instalacao date,
    hipotese character varying(255),
    imprevistos_ocorridos character varying(255),
    local_coleta character varying(255),
    localizacao_parcela character varying(255),
    nome character varying(255),
    objetivo character varying(255),
    parcelas_perdidas character varying(255),
    quant_atividade_experimental integer,
    quant_blocos integer,
    quant_fatores integer,
```

```
    quant_repeticao integer,
    quant_tratamentos integer,
    quant_variavel_resposta integer,
    temperatura_media character varying(255),
    tipo_material character varying(255),
    total_parcelas integer,
    unidade_experimental character varying(255),
    id_estatistico integer,
    id_orientador integer,
    id_responsavel integer,
    id_tecnico_instalaor integer,
    delineamento_experimental character varying(50)
);
ALTER TABLE expdados.experimento OWNER TO postgres;
--
-- Name: experimento_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE experimento_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.experimento_id_seq OWNER TO postgres;
--
-- Name: experimento_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE experimento_id_seq OWNED BY experimento.id;
--
-- Name: fator; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE fator (
    id integer NOT NULL,
    nome character varying(255),
    quant_niveis integer,
    id_experimento integer
);
ALTER TABLE expdados.fator OWNER TO postgres;
--
-- Name: fator_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE fator_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.fator_id_seq OWNER TO postgres;
--
-- Name: fator_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE fator_id_seq OWNED BY fator.id;
--
-- Name: planilha_dados; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE planilha_dados (
    id integer NOT NULL,
    caminho character varying(255),
```

```
    data_criacao date,
    descricao character varying(255),
    nome character varying(255),
    tamanho integer,
    id_experimento integer
);
ALTER TABLE expdados.planilha_dados OWNER TO postgres;
--
-- Name: planilha_dados_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE planilha_dados_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.planilha_dados_id_seq OWNER TO postgres;
--
-- Name: planilha_dados_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE planilha_dados_id_seq OWNED BY planilha_dados.id;
--
-- Name: script; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE script (
    id integer NOT NULL,
    caminho character varying(255),
    data_criacao date,
    descricao character varying(255),
    nome character varying(255),
    tamanho integer,
    id_experimento integer
);
ALTER TABLE expdados.script OWNER TO postgres;

--
-- Name: script_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE script_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.script_id_seq OWNER TO postgres;
--
-- Name: script_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE script_id_seq OWNED BY script.id;
--
-- Name: trabalho_cientifico; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE trabalho_cientifico (
    id integer NOT NULL,
    link character varying(255),
    titulo character varying(255),
    objetivo_trabalho character varying(255),
    tipo_trabalho character varying(255),
```

```
    veiculo character varying(255),
    id_experimento integer,
    caminho character varying(255),
    tamanho integer,
    data_criacao date,
    nome_arquivo character varying(255)
);
ALTER TABLE expdados.trabalho_cientifico OWNER TO postgres;
--
-- Name: trabalho_cientifico_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE trabalho_cientifico_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.trabalho_cientifico_id_seq OWNER TO postgres;
--
-- Name: trabalho_cientifico_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE trabalho_cientifico_id_seq OWNED BY trabalho_cientifico.id;
--
-- Name: usuario; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE usuario (
    id integer NOT NULL,
    cpf character varying(255),
    email character varying(255),
    instituicao character varying(255),
    matr character varying(255),
    nome character varying(255),
    perfil_acesso_sisgexp character varying(255),
    tel character varying(255)
);
ALTER TABLE expdados.usuario OWNER TO postgres;
--
-- Name: usuario_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE usuario_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.usuario_id_seq OWNER TO postgres;
--
-- Name: usuario_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE usuario_id_seq OWNED BY usuario.id;
--
-- Name: variavel_resposta; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE variavel_resposta (
    id integer NOT NULL,
    nome character varying(255),
    id_experimento integer
);
```

```
ALTER TABLE expdados.variavel_resposta OWNER TO postgres;
--
-- Name: variavel_resposta_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE variavel_resposta_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.variavel_resposta_id_seq OWNER TO postgres;
--
-- Name: variavel_resposta_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE variavel_resposta_id_seq OWNED BY variavel_resposta.id;
--
-- Name: wf_script; Type: TABLE; Schema: expdados; Owner: postgres; Tablespace:
--
CREATE TABLE wf_script (
    id integer NOT NULL,
    data_exec timestamp without time zone,
    id_script integer,
    id_workflow_exec integer
);
ALTER TABLE expdados.wf_script OWNER TO postgres;

--
-- Name: wf_script_id_seq; Type: SEQUENCE; Schema: expdados; Owner: postgres
--
CREATE SEQUENCE wf_script_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE expdados.wf_script_id_seq OWNER TO postgres;
--
-- Name: wf_script_id_seq; Type: SEQUENCE OWNED BY; Schema: expdados; Owner: postgres
--
ALTER SEQUENCE wf_script_id_seq OWNED BY wf_script.id;
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY analise_estatistica ALTER COLUMN id SET DEFAULT
nextval('analise_estatistica_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY atividade_experimental ALTER COLUMN id SET DEFAULT
nextval('atividade_experimental_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY delineamento_experimental ALTER COLUMN id SET DEFAULT
nextval('delineamento_experimental_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
```

```
ALTER TABLE ONLY experimento ALTER COLUMN id SET DEFAULT
nextval('experimento_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY fator ALTER COLUMN id SET DEFAULT nextval('fator_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY planilha_dados ALTER COLUMN id SET DEFAULT
nextval('planilha_dados_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY script ALTER COLUMN id SET DEFAULT nextval('script_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY trabalho_cientifico ALTER COLUMN id SET DEFAULT
nextval('trabalho_cientifico_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY usuario ALTER COLUMN id SET DEFAULT nextval('usuario_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY variavel_resposta ALTER COLUMN id SET DEFAULT
nextval('variavel_resposta_id_seq'::regclass);
--
-- Name: id; Type: DEFAULT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY wf_script ALTER COLUMN id SET DEFAULT nextval('wf_script_id_seq'::regclass);
--
-- Name: analise_estatistica_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY analise_estatistica
    ADD CONSTRAINT analise_estatistica_pkey PRIMARY KEY (id);
--
-- Name: atividade_experimental_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY atividade_experimental
    ADD CONSTRAINT atividade_experimental_pkey PRIMARY KEY (id);
--
-- Name: experimento_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY experimento
    ADD CONSTRAINT experimento_pkey PRIMARY KEY (id);
--
-- Name: fator_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY fator
    ADD CONSTRAINT fator_pkey PRIMARY KEY (id);
--
-- Name: pk_id_delineamento_exp; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY delineamento_experimental
    ADD CONSTRAINT pk_id_delineamento_exp PRIMARY KEY (id);
```

```
--
-- Name: pk_id_wfscript; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY wf_script
   ADD CONSTRAINT pk_id_wfscript PRIMARY KEY (id);
--
-- Name: planilha_dados_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY planilha_dados
   ADD CONSTRAINT planilha_dados_pkey PRIMARY KEY (id);
--
-- Name: script_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY script
   ADD CONSTRAINT script_pkey PRIMARY KEY (id);
--
-- Name: trabalho_cientifico_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY trabalho_cientifico
   ADD CONSTRAINT trabalho_cientifico_pkey PRIMARY KEY (id);
--
-- Name: usuario_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY usuario
   ADD CONSTRAINT usuario_pkey PRIMARY KEY (id);
--
-- Name: variavel_resposta_pkey; Type: CONSTRAINT; Schema: expdados; Owner: postgres; Tablespace:
--
ALTER TABLE ONLY variavel_resposta
   ADD CONSTRAINT variavel_resposta_pkey PRIMARY KEY (id);
--
-- Name: fk_experimento_id_estatistico; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY experimento
   ADD CONSTRAINT fk_experimento_id_estatistico FOREIGN KEY (id_estatistico) REFERENCES
usuario(id);
--
-- Name: fk_experimento_id_orientador; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY experimento
   ADD CONSTRAINT fk_experimento_id_orientador FOREIGN KEY (id_orientador) REFERENCES
usuario(id);
--
-- Name: fk_experimento_id_responsavel; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY experimento
   ADD CONSTRAINT fk_experimento_id_responsavel FOREIGN KEY (id_responsavel) REFERENCES
usuario(id);
--
-- Name: fk_experimento_id_tecnico_instalaor; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY experimento
   ADD CONSTRAINT fk_experimento_id_tecnico_instalaor FOREIGN KEY (id_tecnico_instalaor)
REFERENCES usuario(id);
--
-- Name: fk_fator_id_experimento; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY fator
```

```
    ADD CONSTRAINT fk_fator_id_experimento FOREIGN KEY (id_experimento) REFERENCES
experimento(id);
--
-- Name: fk_id_experimento_atividadeexperimental; Type: FK CONSTRAINT; Schema: expdados; Owner:
postgres
--
ALTER TABLE ONLY atividade_experimental
    ADD CONSTRAINT fk_id_experimento_atividadeexperimental FOREIGN KEY (id_experimento)
REFERENCES experimento(id);
--
-- Name: fk_planilha_dados_id_experimento; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY planilha_dados
    ADD CONSTRAINT fk_planilha_dados_id_experimento FOREIGN KEY (id_experimento) REFERENCES
experimento(id);
--
-- Name: fk_script_id_experimento; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY script
    ADD CONSTRAINT fk_script_id_experimento FOREIGN KEY (id_experimento) REFERENCES
experimento(id);
--
-- Name: fk_trabalho_cientifico_id_experimento; Type: FK CONSTRAINT; Schema: expdados; Owner:
postgres
--
ALTER TABLE ONLY trabalho_cientifico
    ADD CONSTRAINT fk_trabalho_cientifico_id_experimento FOREIGN KEY (id_experimento)
REFERENCES experimento(id);
--
-- Name: fk_variavel_resposta_id_experimento; Type: FK CONSTRAINT; Schema: expdados; Owner: postgres
--
ALTER TABLE ONLY variavel_resposta
    ADD CONSTRAINT fk_variavel_resposta_id_experimento FOREIGN KEY (id_experimento)
REFERENCES experimento(id);
--
-- PostgreSQL database dump complete
--
```