UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Tássio Ferenzini Martins Sirqueira

E-SECO ProVersion: Uma arquitetura para Manutenção e Evolução de *Workflows* Científicos

Juiz de Fora

2016

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Tássio Ferenzini Martins Sirqueira

E-SECO ProVersion: Uma arquitetura para Manutenção e Evolução de *Workflows* Científicos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Regina Maria Maciel Braga Villela
Coorientador: Marco Antônio Pereira Araújo

Juiz de Fora

2016

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Sirqueira, Tassio Ferenzini Martins.
E-SECO ProVersion : Uma arquitetura para Manutenção e Evolução de Workflows Científicos / Tassio Ferenzini Martins Sirqueira. -- 2016.
151 f.

Orientadora: Regina Maria Maciel Braga
Coorientador: Marco Antônio Pereira Araújo
Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós Graduação em Ciência da Computação, 2016.

1. Workflow Científico. 2. Proveniência de Dados. 3. Manutenção. 4. Evolução. I. Braga, Regina Maria Maciel, orient. II. Araújo, Marco Antônio Pereira, coorient. III. Título.

Tássio Ferenzini Martins Sirqueira

E-SECO ProVersion: Uma arquitetura para Manutenção e Evolução de *Workflows* Científicos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 12 de Julho de 2016.

BANCA EXAMINADORA

Profa. D.Sc. Regina Maria Maciel Braga Villela - Orientadora
Universidade Federal de Juiz de Fora

Prof. D.Sc. Marco Antônio Pereira Araújo - Coorientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Alcione de Paiva Oliveira
Universidade Federal de Viçosa

Prof. D.Sc. José Maria Nazar David
Universidade Federal de Juiz de Fora

A Deus em primeiro lugar. A
minha família, noiva,
orientadores e amigos pelo apoio.

AGRADECIMENTOS

A minha mãe Ivana, meu padrasto Moacir e minha sobrinha Anna Beatriz por todo o apoio e esforços despendidos à me ajudar.

A meus avós Rubens (em memória) e Apparecida por todo incentivo e ajuda ao longo desta caminhada.

A minha noiva Jéssica que acompanhou tudo de perto, sempre me apoiando e me incentivando a melhorar.

A minha orientadora Regina Braga por todo o apoio e todas as colaborações durante esses anos juntos.

Ao meu coorientador Marco Antônio Araújo por todo o apoio ao longo da minha carreira acadêmica e profissional.

Aos amigos do *Forever Alone*, Humberto, Marcos e Wellington e aos agregados (Gabriella e Camila) pelas incansáveis discussões e por tornarem o ambiente mais agradável.

Aos professores do PGCC pelas imensas contribuições ao meu aprendizado.

A todos do núcleo de informática do Instituto Federal de Educação Ciência e Tecnologia do Campus de Juiz de Fora pelos incentivos e apoios na continuação de meus estudos.

E por fim, não menos importantes, aos meus amigos que sempre estiveram comigo ajudando e apoiando minhas conquistas.

'Tudo o que é seu encontrará
uma maneira de chegar até você'
Chico Xavier

RESUMO

Um ecossistema de software científico, além de outras funcionalidades, busca integrar todas as etapas de um experimento, e comumente utiliza *workflows* científicos para a resolução de problemas complexos. Toda modificação ocorrida em um experimento deve ser propagada para os *workflows* associados, os quais devem ser mantidos e evoluídos para o prosseguimento com sucesso da pesquisa. Um das forma de garantir este controle é através da gerência de configuração.

Para que ela possa ser utilizada, é importante o armazenamento dos dados de execução e modelagem do experimento e *workflows* associados. Neste trabalho, utilizamos conceitos e modelos relacionados à proveniência de dados para o armazenamento e consulta destes dados. O uso da proveniência de dados traz alguns benefícios neste armazenamento e consulta, conforme veremos nesta dissertação.

Assim, nesse trabalho é proposta uma arquitetura para gerenciar a evolução e manutenção de experimentos e *workflows* científicos, denominada E-SECO ProVersion. A motivação para a especificação e implementação da arquitetura veio a partir da realização de uma revisão sistemática e de um estudo para verificar características de manutenção e evolução em repositórios de *workflows* existentes. A partir destas análises, as principais funcionalidades da arquitetura foram definidas e detalhadas. Além disso, um roteiro com diretrizes de uso e provas de conceito utilizando *workflows* extraídos do repositório myExperiment foram apresentados, com o objetivo de avaliar a aplicabilidade da arquitetura.

Palavras-chave: Workflow Científico. Manutenção. Evolução. Proveniência de Dados.

ABSTRACT

A scientific software ecosystem, in addition to other features, seeks to integrate all stages of an experiment, and commonly used scientific workflows to solve complex problems. Any changes that occurred in an experiment must be propagated to the associated workflows, which must be maintained and evolved for further successful research. One of the way to ensure this control is through configuration management.

So that it can be used, it is important the storage of performance data and modeling of the experiment and associated workflows. In this study, we use the concepts and models related to the source of data for storage and retrieval of this data. Use the data source brings some advantages in storage and query, as we will see in this dissertation.

Thus, this paper proposes an architecture to manage the development and maintenance of scientific experiments and workflows, called E-SECO ProVersion. The motivation for the specification and implementation of architecture came from the realization of a systematic review and a study to check maintenance characteristics and evolution in existing workflows repositories. From these analyzes, the main features of the architecture are defined and detailed. In addition, a roadmap with usage guidelines and proofs of concept using workflows extracted from myExperiment repository were presented in order to evaluate the applicability of architecture.

Keywords: Scientific Workflow. Maintenance. Evolution. Data Provenance.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface

GQM - Goal/Question/Metric

LPS - Linha De Produtos De Software

LPSC - Linha De Produtos De Software Científico

OPM - Open Provenance Model

OWL - Web Ontology Language

P2P - Ponto A Ponto ou Peer To Peer

PICOC - População, Intervenção, Comparação, Saída(Output), Contexto

PoC - Prova De Conceito

SGBD - Sistema Gerenciador De Banco De Dados

SGWfC - Sistema Gerenciador De Workflow Científico

UFJF - Universidade Federal de Juiz de Fora

W3C - World Wide Web Consortium

WFMC - Workfow Management Coalition

XLS - Formato De Arquivos Do Microsoft Excel

XML - eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

Processos de *workflow* são fortemente utilizados em aplicações de workgroup, ou sistemas colaborativos (CRUZ, 2006) . Segundo CRUZ (2006), no modelo de *workflow* a ênfase é dada ao processo, onde regras orientam a execução de cada tarefa, permitindo um nível de detalhamento e precisão não existente em nenhum outro modelo. Neste contexto, um *workflow* científico é um modelo ou template que representa uma sequência de atividades científicas implementadas por ferramentas, a fim de alcançar um determinado objetivo (DEELMAN *et al.*, 2009) . Este modelo deve permitir a interpretação e execução por parte dos chamados Sistemas Gerenciadores de *Workflows* Científicos (SGWfC).

Workflows científicos diferem-se de *workflows* aplicados a sistemas de negócios por possuírem características que acrescentam-lhes maior complexidade, tais como: (i) fluxos com um grande número de etapas; (ii) volatilidade dos fluxos, passíveis de alterações frequentes; e (iii) necessidade de parametrização para um grande número de tarefas (NARDI, 2009) .

Um experimento científico é definido como uma série de operações de análise (atividades) ligadas entre si (GOBLE *et al.*, 2010) , que podem ser executadas utilizando um ou mais *workflows*, de acordo com a complexidade do experimento. Porém, em geral, os SGWfC não apoiam uma documentação mais detalhada do experimento como um todo, ficando esse conhecimento retido somente ao pesquisador responsável pelo mesmo (PEREIRA; ARAÚJO; TRAVASSOS, 2009) .

Para auxiliar na tarefa de gerenciamento dos experimentos e dos *workflows* científicos, este trabalho aborda o controle da evolução e manutenção de experimentos científicos, utilizando dados de proveniência, no contexto da plataforma de ecossistema E-SECO (FREITAS *et al.*, 2015) .

O E-SECO é uma plataforma de ecossistema aberta, utilizada para apoiar a especificação e a condução de experimentos científicos. A arquitetura E-SECO ProVersion pode ser considerada como um aplicativo agregado à plataforma E-SECO, para suporte a manutenção e evolução de experimentos e *workflows* científicos, fornecendo, de maneira automatizada, informações estratégicas relacionadas ao experimento e *workflows*, para que

os pesquisadores possam tomar decisões ou obter conhecimento estratégico relacionado aos experimentos.

1.1 DEFINIÇÃO DO PROBLEMA

Conforme dito acima, um *workflow* científico é geralmente modelado e executado utilizando um SGWfC (ZHAO *et al.*, 2004) . Entretanto, os SGWfC não apoiam por completo a modelagem de *workflows* (MCPHILLIPS *et al.*, 2009) , limitando-se a gerenciar a execução de um *workflow* científico de forma isolada ao experimento do qual faz parte, ou seja, não se tem controle sobre o experimento conduzido e seus *workflows* associados.

Neste contexto, de acordo com Hasan, Sion e Winslett (2007), faz-se necessário o uso de ferramentas independentes do SGWfC para gerenciar o experimento e os *workflows* associados, analisando inclusive os dados de proveniência. Assim, para representar e apoiar o desenvolvimento de um experimento científico, é necessário o registro dos *workflows* associados e suas variações, visto que os mesmos podem ser modificados no decorrer da pesquisa (MATTOSO *et al.*, 2009) .

Como abordado por Marinho *et al.* (2012), um dos problemas de processos de experimentação que utilizam *workflows*, é a perda de conhecimento do pesquisador sobre o experimento como um todo, devido à delegação das tarefas para o computador que, geralmente, realiza ações isoladas em relação aos *workflows* e que não são documentadas adequadamente, conforme já mencionado. Esse fato causa um descontrole de informação, haja vista que o pesquisador desconhece a origem dos dados que foram gerados na execução do experimento, através de qual *workflow* foi gerado e a versão do *workflow* utilizada. O desconhecimento de tais informações dificulta a análise quanto à validade das mesmas.

A derivação de *workflows* científicos durante a composição de um experimento favorece o uso da abordagem descendente para a especificação de novos *workflows*, (MARINHO *et al.*, 2012) . Essa abordagem permite ao pesquisador a composição de novos *workflows* com base na reutilização de modelos existentes. Para auxiliar na reutilização, todos os dados sobre um experimento e os *workflows* a ele vinculados devem ser registrados na gerência de configuração do experimento, que conforme (ESTUBLIER, 2000) deve apresentar

algumas características, as quais são especificadas a seguir:

- Repositório com acesso controlado, onde os *workflows* possam ser armazenados e onde se possa registrar quais versões do *workflow* são estáveis e quais estão em desenvolvimento ou necessitando manutenção;
- Mecanismos para representar ou armazenar as versões das atividades utilizadas durante a composição de um determinado *workflow*;
- Adição de funcionalidades para apoio aos chamados laboratórios colaborativos (OLSON, 2009) , que apoiem tanto o desenvolvimento de um experimento, de forma distribuída e colaborativa, como também os *workflows* associados, fazendo a gerência inclusive da publicação dos mesmos junto aos repositórios distribuídos;
- Gestão das atividades, ou seja, quais atividades estão sendo mais utilizadas e quais *workflows* seriam afetados durante uma possível modificação de uma atividade.

Com isto, o problema a ser tratado na solução proposta é a gerência dos experimentos e dos *workflows* associados, utilizando como base o controle dos dados de proveniência gerados durante a composição e execução dos *workflows*, bem como as versões dos *workflows* utilizadas no experimento. Essas informações auxiliam o pesquisador a entender e a gerenciar como os experimentos devem ser mantidos e evoluídos no decorrer da pesquisa, assim como os seus *workflows*.

1.2 OBJETIVOS

Este trabalho tem como objetivo identificar informações sobre manutenção e evolução de experimentos e *workflows* científicos e, com base nestas informações, auxiliar os pesquisadores nas análises dos dados dos experimentos, pois cada fase do ciclo do experimento científico apresenta tarefas específicas, onde a cada modificação na forma de execução dessas tarefas, surgem novas versões de *workflows* e consequentemente, influenciam o experimento. Para controlar essas manutenções e evoluções, pode-se utilizar princípios de "Gerência de Configuração", conforme dito acima.

Para que seja possível a gerência de configuração de um experimento, é necessária a coleta de dados sobre o mesmo e das atividades associadas. Uma das formas de coletar os dados do experimento, e dos *workflows*, é através dos dados de proveniência (GASPAR *et al.*, 2015) gerados durante a composição (modelagem) e execução do experimento.

1.3 ABORDAGEM PROPOSTA

Como proposta de solução para o problema de gerenciamento da evolução e das manutenções de experimentos e *workflows* relacionados a um ciclo de experimentação, a proposta deste trabalho especifica uma arquitetura, denominada E-SECO ProVersion, que trata a coleta dos dados de modelagem e execução do *workflow* através de um modelo de proveniência de dados, no contexto da plataforma de ecossistemas E-SECO.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em cinco capítulos além desta introdução, conforme Figura 1.1. O capítulo 2 apresenta a fundamentação teórica do trabalho, abordando conceitos de manutenção e evolução de software, ciclo de vida de experimentos científicos, modelos de proveniência de dados, ontologia e alguns dos principais SGWfC existentes. O capítulo 3 apresenta uma revisão sistemática e os trabalhos relacionados a proposta desenvolvida neste trabalho. O capítulo 4 apresenta a proposta deste trabalho, descrevendo a arquitetura da solução, denominada E-SECO ProVersion, a modelagem de dados para a captura da proveniência, de acordo com o modelo de proveniência PROV, a modelagem da ontologia para extração de conhecimento implícito, denominada PROV-OEXT, e o uso destas tecnologias para o gerenciamento dos experimentos e dos *workflows* associados. Já no capítulo 5 são apresentadas algumas diretrizes de uso, com uma análise dos repositórios de *workflows* disponíveis, uma discussão sobre as informações disponíveis acerca destes *workflows*, considerando características de evolução e manutenção ao longo do ciclo de vida. Além disso, um roteiro de uso da arquitetura e uma prova de conceitos com *workflows* do myExperiment também são detalhados. Por fim, o capítulo 6 apresenta as considerações finais, bem como sugestões de trabalhos futuros.

Figura 1.1: Estrutura da dissertação

2 PRESSUPOSTOS TEÓRICOS

Este capítulo aborda os principais temas de pesquisa relacionados a esta dissertação, incluindo experimentação científica, proveniência de dados, ontologias, manutenção e evolução de software. Conceitos relacionados a estes temas são importantes para embasar o leitor para o entendimento dos próximos capítulos.

2.1 EXPERIMENTAÇÃO CIENTÍFICA

Com a evolução da informática e o surgimento da Internet, a ciência passou a explorar novas possibilidades de experimentação científica (OGASAWARA *et al.*, 2008) , em ambientes multidisciplinares. Como abordado pelo relatório do Seminário "Grandes Desafios de Pesquisa em Computação no Brasil: 2006 - 2016", promovido pela Sociedade Brasileira de Computação (SBC), o uso de sistemas computacionais torna-se necessário para possibilitar que pesquisadores em outros domínios do conhecimento possam investigar problemas que, até recentemente, não podiam ser tratados. Neste contexto de uso de recursos computacionais para a experimentação científica, surge o termo e-ciência ou e-Science (HEY; TREFETHEN, 2003) .

O termo e-Science foi definido por Hey e Trefethen (2003) como uma colaboração global de áreas-chave da ciência junto com a geração de uma infraestrutura capaz de suportá-la. No entanto, e-Science vai além deste conceito, pois caracteriza-se pelo acesso a uma vasta coleções de dados, uso computacional em grande escala, heterogeneidade de recursos, reusabilidade e uso de *workflows* científicos (SILVA; BRAGA; CAMPOS, 2012) .

Como abordado por Siqueira *et al.* (2008), mesmo com a evolução da ciência e do modo de se fazer pesquisa, o conhecimento empírico ainda encontra-se presente na tripla "tentativa-erro-repetição". Apesar de se ter, atualmente, uma formulação mais rigorosa, as falhas ou erros fazem parte da ciência, e é necessária a sua documentação para auxiliar em estudos futuros, prevenindo outros pesquisadores de cometerem os mesmos erros já vistos em pesquisas anteriores. Isso reforça a necessidade de gerenciamento dos experimentos

científicos.

Para que seja possível o gerenciamento de experimentos científicos, é importante o controle da modelagem e criação dos *workflows*, além da gerência do ciclo de vida do experimento. Nas subseções seguintes, iremos detalhar alguns destes conceitos.

2.1.1 *WORKFLOWS* CIENTÍFICOS

Um *workflow* científico pode ser caracterizado como sendo a composição de um conjunto de atividades científicas, que devem ser executadas em uma ordem pré-determinada, com o objetivo de automatizar um dado experimento científico. Taylor *et al.* (2014) caracterizam um *workflow* como a execução automatizada de procedimentos até então realizados de forma manual, através do qual os objetos de análise dos experimentos são usualmente processados por simulações computacionais e analisados via técnicas de visualização (DEELMAN *et al.*, 2009) .

Um experimento científico por sua vez é definido como uma série de operações de análise ligadas entre si, no qual estas ligações são modeladas e executadas através de *workflows* científicos (GOBLE *et al.*, 2010) . Os *workflows* também consistem de diversos componentes que, de acordo com Pacheco (apud MARSHAK, 1995), são:

- Tarefas: composto por diferentes tarefas ou atividades, que devem ser cumpridas para se atingir um objetivo;
- Pessoas: pessoas específicas ou entidades automatizadas (realizando tarefas de pessoas) que desempenham as atividades em uma ordem determinada;
- Ferramentas: o sistema de *workflow* não desempenha todas as atividades em si. Muitas vezes faz uso de sistemas computacionais específicos e dedicados para executar determinadas tarefas;
- Dados: são as informações acessadas pelas ferramentas para realizar as tarefas.

A união destes componentes permite automatizar um processo por completo ou parcialmente, e essa automação deve representar um formato compreensível por uma máquina (SILVA; SOARES; BRAGA, 2006) . Embora a modelagem e execução de *workflow* sejam

relativamente novas, muitos dos seus conceitos já existem há algum tempo e têm sido aplicados a diversas áreas (SILVA; SOARES; BRAGA, 2006 apud SATLER, 2004).

Com base nessa perspectiva, alguns autores (MEDEIROS *et al.*, 2005; GIL *et al.*, 2007) propõem que um *workflow* deva inicialmente ser concebido em alto nível de abstração, com o intuito de definir o objetivo, atividades e escopo, e só então, definir a tecnologia que será empregada, isso é, com uso de Sistemas de Gerenciamento de *Workflows* Científicos (SGWfC)[1]. Nesse caso, o nível mais abstrato é ligado à definição do comportamento do *workflow*, sendo denominado *workflow* abstrato. Já o nível concreto é ligado aos recursos computacionais necessários à execução do *workflow* científico, estando esse pronto para execução em um SGWfC, sendo denominado *workflow* concreto (MATTOSO *et al.*, 2009) .

Sistemas de Gerenciamento de *Workflows* Científicos (SGWfCs) foram desenvolvidos (DEELMAN *et al.*, 2009) com o objetivo de propiciar a orquestração de algoritmos, tirando partido do processamento paralelo e distribuído, bancos de dados, inteligência artificial, dentre outros, construindo assim um arcabouço para experimentação através de simulação Taylor *et al.* (2014). Um SGWfC é um sistema que define e organiza a execução de *workflow* pelo uso de software, sendo capaz de interpretar a definição de um processo, interagir com os participantes e invocar o uso de ferramentas e aplicações quando necessário (SILVA; SOARES; BRAGA, 2006) . Assim, os SGWfCs provêem um conjunto de ferramentas de software, para o apoio a definição e execução de *workflows*.

Um *workflow* pode se comparar a um processo de desenvolvimento de software, no qual deve ser bem pensado e planejado, levando em consideração o seu ciclo de vida, uma vez que sofrerá diversas modificações ao longo do projeto. Todavia, essa tecnologia gera novas questões associadas à especificação, modelagem e reutilização em estudos experimentais (MATTOSO *et al.*, 2009) , levando a uma necessidade de aprimorar a organização das informações sobre o *workflow*.

A Workfow Management Coalition (WfMC) é uma organização internacional que tem como objetivo promover o uso de *workflows* através de padrões de software relacionados à terminologia, interoperabilidade e conectividade. Baseado nessa ideia, a WfMC propôs um modelo, conforme apresentado na Figura 2.1.

[1]Este conceito será detalhado mais adiante no texto.

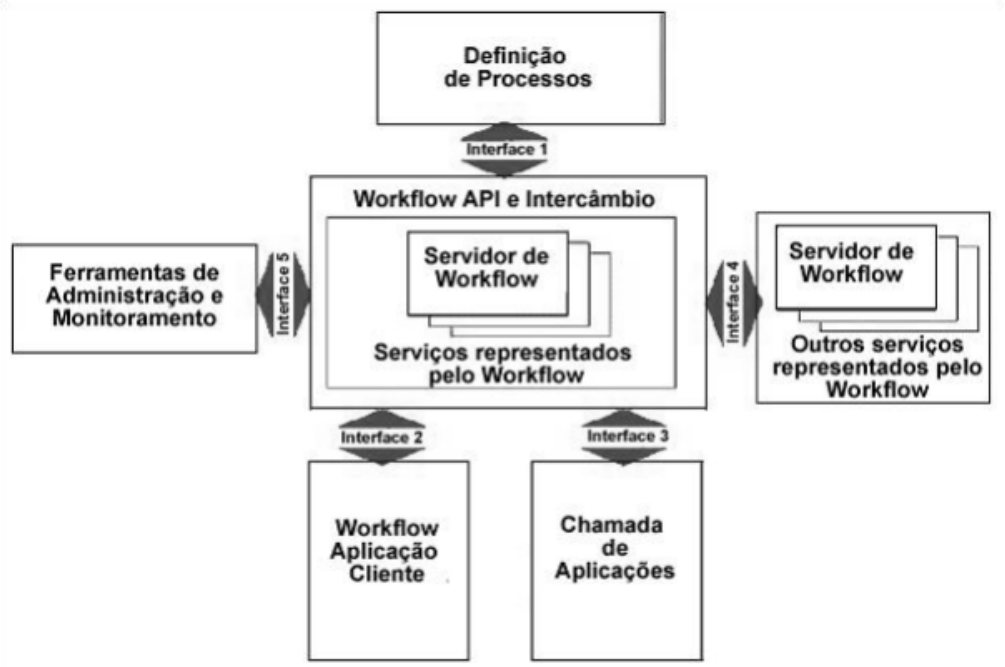


Figura 2.1: Modelo padrão da WfMC (CRUZ, 2006) .

O modelo proposto pela WfMC define que devem existir servidores de *workflow* os quais devem possuir 5 interfaces de comunicação, para administração e monitoramento, aplicação com cliente, instanciação, integração com outros servidores de *workflows* e, por fim, uma interface de definição dos processos empregados no *workflow*.

A Interface 1 objetiva apoiar a comunicação entre o módulo de definição e criação de processo com o motor de execução do *workflow*, permitindo entender e executar os comandos passados. A Interface 2 trata da portabilidade e reuso dos clientes em diferentes sistemas de *workflows*, fundamentado no quesito de independência entre as partes. A Interface 3 traz o conceito de vocabulário, permitindo que o sistema de *workflow* se integre em diferentes produtos e sistemas sem perder a independência. Na Interface 4, o modelo trata da interoperabilidade entre os diferentes sistemas de *workflow*. Essa interoperabilidade pode ser em 3 níveis: I - motor do *workflow*, II - serviços e III - sistemas de gerenciamento do *workflow*, conforme a Figura 2.2. Por fim, na interface 5 é tratada a supervisão, gerenciamento e controle do ambiente no qual o sistema de *workflow* é executado.

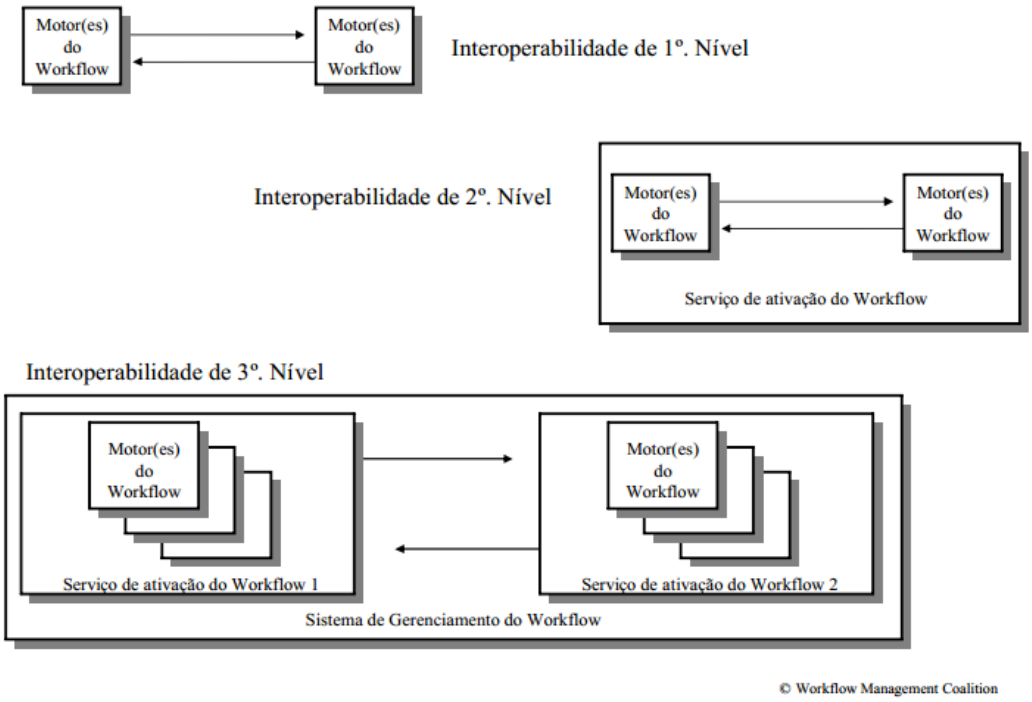


Figura 2.2: Níveis da Interface 4 do modelo do WfMC (CRUZ, 2006) .

O modelo da WfMC também define as ligações entre os objetos e recursos que fazem parte do *workflow*. Mas a concepção de um *workflow* não é algo trivial, demanda entender o problema a ser tratado e identificar formas possíveis de tratá-lo. Todos esses passos precisam ser planejados. O modelo concebido pela WfMC persegue algumas virtudes que são:

- Abstrato: a especificação deve ser independente de qualquer software, permitindo que a automatização possa contemplar qualquer tipo de processo;
- Independente: o modelo deve permitir especificar qualquer tipo de *workflow* que possa ser executado em qualquer sistema de gerenciamento de *workflow*, não sendo assim um modelo proprietário;
- Vocabulário: o modelo WfMC estimula padronização de um vocabulário. O modelo de *workflow* do WfMC, desde sua concepção, sofreu e ainda sofre atualizações e melhorias (CRUZ, 2006) . Embora cada uma das interfaces apareçam separadas dentro do modelo do WfMC, estas agem em conjunto por serem executadas de forma interdependente. Esse conceito define o ciclo de vida do *workflow*, conforme a Figura 2.3.

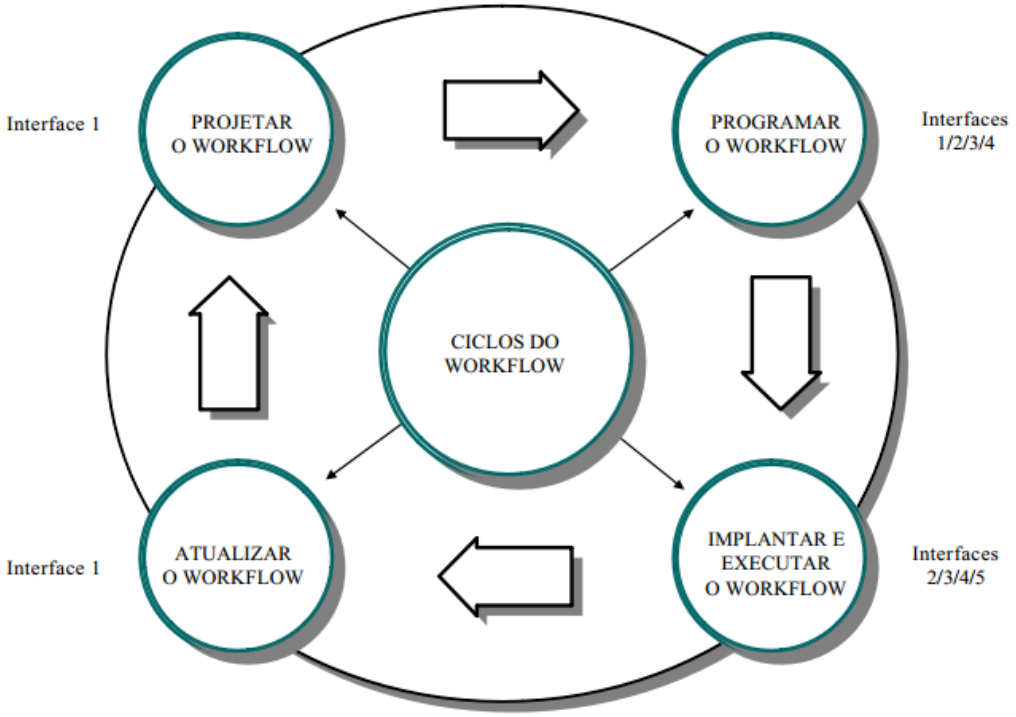


Figura 2.3: Ciclo de vida do *workflow* de acordo com WfMC (CRUZ, 2006) .

A WfMC preza o uso de padrões de software para aplicações que utilizam *workflows*, desta forma a interoperabilidade, conectividade, administração e monitoramento dos *workflows*, bem como dos dados processados se tornariam mais fáceis de serem gerenciados o que contribuiria para o reuso destes. No entanto, considerando o contexto de experimentação científica, a modelagem e execução de *workflows* não seguem estritamente os padrões da WfMC, e cada SGWfC segue seu próprio padrão, o que leva a necessidade de ferramentas externas para acompanhar e gerenciar as pesquisas e *workflows* vinculados.

Com o objetivo de caracterizar a área de pesquisa que envolve experimentos e *workflows* científicos, buscou-se identificar métodos, ferramentas ou modelos que tratassem do uso de *workflows* e a ligação com os experimentos. O resultado dessa busca na literatura técnica resultou na lista de aplicações dispostas a seguir:

- BioSide (BIGARET; MEYER, 2008) : é uma ferramenta com foco na construção de *workflows* em bioinformática e que permite o acesso a outros programas para auxiliar na execução do experimento, contudo, não gerência o experimento;
- E-BioFlow (INGOWASSINK; VET, 2008) : sistema de *workflow* que trata princi-

palmente a interação com o usuário e questões de proveniência dos dados, mas não se preocupa com as manutenções e evoluções do mesmo;

- Galaxy (GOECKS *et al.*, 2010) : plataforma aberta baseada na Web, para pesquisas intensivas em dados biomédicos, que como o E-BioFlow, não trata da manutenção e evolução dos experimento e *workflows* científicos;

- Kepler (ALTINTAS *et al.*, 2004) : um sistema de *workflow* com fluxo de dados que é usado no domínio de ecologia, geologia e bioinformática, possui um plug-in para tratar a proveniência dos dados gerados na execução do *workflow*, mas não os associa ao experimento ao qual faz parte;

- Lims (VOEGELE *et al.*, 2007) : possui elementos de um sistema de *workflow*, mas é projetado principalmente como um sistema de informação de laboratório de análise de dados experimentais, usando "G" que é uma linguagem orientada a fluxo de dados;

- Pegasus (DEELMAN *et al.*, 2004) : engloba um conjunto de tecnologias baseadas em *workflow*, permite a modelagem, execução e acompanhamento dos *workflows* em execução, mas não vincula os *workflows* aos seus respectivos experimentos;

- Taverna (OINN *et al.*, 2007) : focado principalmente em apoio à comunidade de ciências da vida (biologia, química e médica), usa um modelo orientado a fluxo de dados, permite a capturar os traços de proveniência seguindo o modelo de proveniência PROV e acesso a repositórios de *workflows*, contudo, não trata dos dados do experimento como um todo;

- Triana (TAYLOR *et al.*, 2007) : ferramenta de *workflow* e análise de dados gráficos, incluindo sinal, texto e processamento de imagem. Inclui uma biblioteca de ferramentas e os usuários podem integrar suas próprias ferramentas, Web e Grid Services, mas como os anteriores não se preocupa com os dados do experimento;

- Trident (BARGA *et al.*, 2008) : sistema de *workflow* desenvolvido pela Microsoft Research. Baseia-se no fluxo de trabalho do Windows Foundation (WF), seu foco não é *workflows* científicos;

- VisTrails (FREIRE *et al.*, 2006) : é um sistema de gestão científica de *workflows* e proveniência que fornece suporte para a exploração e visualização de dados. Apesar de tratar a proveniência gerada durante a execução do *workflow*, não se preocupa com os dados do experimento;

- XBaya (SHIRASUNA; GANNON, 2006) : sistema de *workflow* para motores como a ODE e ActiveBPEL. Pode ser usado como um aplicativo independente ou como uma aplicação Java Web, seu foco é *workflows* de negócio.

As ferramentas identificadas na literatura não tratam, de modo geral, dos requisitos de um experimento tais como especificação, verificação das entradas e saídas e dos resultados obtidos, bem como a otimizar dos processos do experimento e dos *workflows*. Nem seguem estritamente as especificações do WfMC. Além disso, foi observado que as ferramentas listadas não possuem um tratamento específico para o ciclo de vida do *workflow*, bem como para o ciclo de vida do experimento científico. O Taverna por exemplo, permite acessar o repositório de *workflow* myExperiment (GOBLE *et al.*, 2010) , entre outros. Mas o mesmo não trata o versionamento do *workflow* e como será distribuído para a comunidade científica, nem trata as informações referentes ao seu uso e do experimento ao qual faz parte.

Assim, somente com o suporte do SGWfCs, no momento da concepção do *workflow* científico, o pesquisador enfrenta uma série de dificuldades. Além disso, muitas vezes essa tarefa é realizada diretamente no nível concreto e de maneira ad hoc, o que pode acarretar em riscos para a pesquisa (GIL *et al.*, 2007; VERDI; ELLIS; GRYK, 2007) . Como os SGWfC se limitam a gerenciar a execução de um *workflow* científico de forma isolada do experimento ao qual faz parte, para representar e apoiar o desenvolvimento do experimento científico, é necessário o registro das variações dos *workflows* associados a um experimento, visto que o mesmo é modificado no decorrer das pesquisas. Essas variações incluem a mudança de dados de entrada, parâmetros, programas ou ainda a combinação delas (OGASAWARA *et al.*, 2008; OLIVEIRA *et al.*, 2008) . Para este controle é necessário o entendimento e gerência do ciclo de vida do *workflow* e do experimento como um todo.

2.1.2 CICLO DE VIDA DE EXPERIMENTOS CIENTÍFICOS

Conforme já dito, o entendimento e a gerência do ciclo de vida de um experimento científico e seus *workflows* associados são partes importantes para o sucesso de um experimento. Assim, descrevemos a seguir as principais características relacionadas ao ciclo de vida tanto de *workflows* quanto de experimentos científicos.

Deelman e Gil (2006) listam as fases de um ciclo de vida de um *workflow* científico como: i) Aprendizado e Análise: visa estabelecer a concepção do problema e o tratamento devido; ii) Projeto e Refinamento: define os recursos necessários e os agentes envolvidos, bem como o fluxo entre eles; iii) Planejamento e Compartilhamento: aborda a construção do modelo e sua disponibilidade para uso; iv) Execução: realiza a execução propriamente dita do modelo. Esses passos foram definidos como o ciclo de vida clássico, conforme a Figura 2.4.

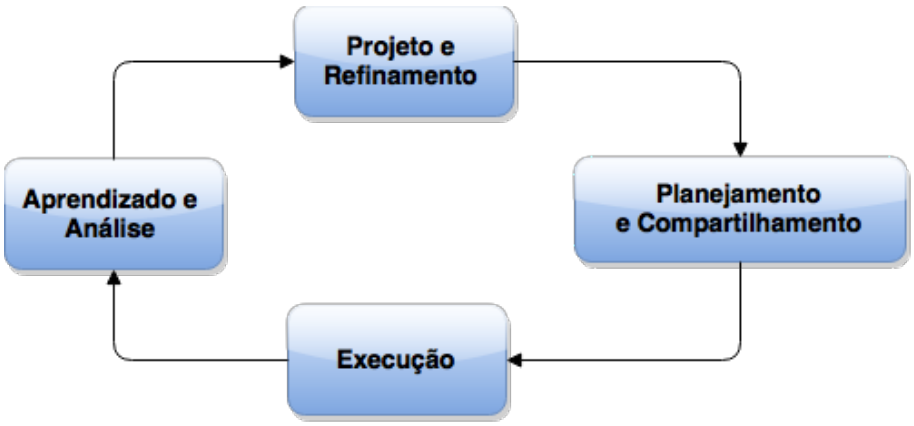


Figura 2.4: Ciclo de vida de um *workflow* científico (DEELMAN; GIL, 2006) .

Com base no ciclo de vida de Deelman e Gil (2006), Holl *et al.* (2014) propuseram a criação de uma nova fase, que é de otimização do *workflow* científico, conforme mostra a Figura 2.5.

Essa nova fase tem como função identificar falhas ou pontos de melhoria no processamento do *workflow* e assim adaptá-lo para torná-lo mais claro e fácil de reutilizar. Neste contexto, Pereira, Araújo e Travassos (2009) definiram um modelo para a concepção de *workflows* abstratos, tendo como base um modelador, que pode ser um pesquisador ou engenheiro de software, responsável pela especificação e modelagem do *workflow*. Esse modelo pode ser visto na Figura 2.6.

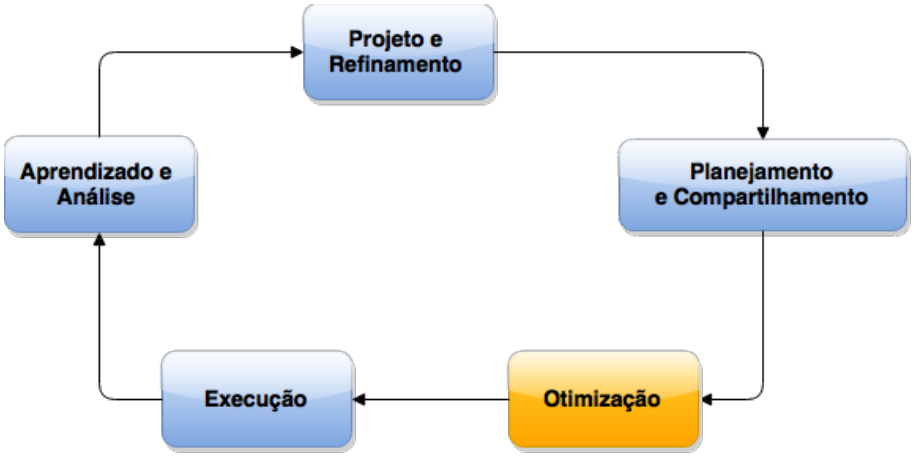


Figura 2.5: Novo modelo de ciclo de vida do *workflow* científico (HOLL *et al.*, 2014) .

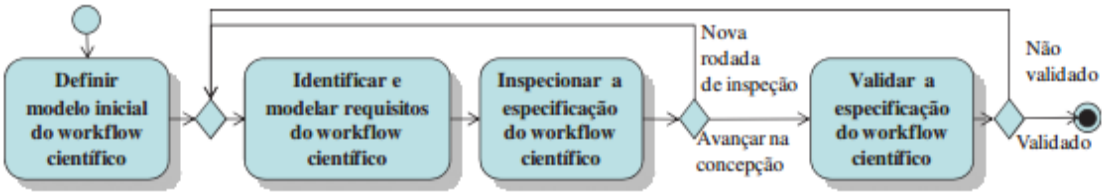


Figura 2.6: Abordagem para Concepção de *Workflows* Abstratos (PEREIRA; ARAÚJO; TRAVASSOS, 2009) .

Inicialmente, realiza-se a tarefa "Definir o modelo inicial do *workflow* científico", onde o modelador concebe o modelo inicial, construindo uma visão global do estudo através da discussão com outros pesquisadores. Após essa tarefa, "Identificar e modelar requisitos do *workflow* científico" é executado. Nela, o modelador cria a especificação do *workflow* abstrato através de reuniões semiestruturadas com os pesquisadores. Os formulários são utilizados como guias nas perguntas da entrevista e o modelo inicial como base para o modelo de *workflow* abstrato (PEREIRA; ARAÚJO; TRAVASSOS, 2009) .

Acreditar que um *workflow* não sofrerá evolução e mudanças no contexto de um experimento pode ser considerado utópico ou ingênuo, pois à medida que novos resultados vão surgindo, a pesquisa vai tomando novos rumos, sendo necessário replanejamento, modificação ou adaptação na forma de execução, ou até mesmo dos recursos externos utilizados, como um serviço *Web* ou *sub-workflows* de pré ou pós processamento. Assim, considera-se de grande importância também o estudo do ciclo de vida do experimento científico.

Oinn *et al.* (2007) definem que os experimentos científicos devem ser divididos em cinco etapas, conforme a Figura 2.7.

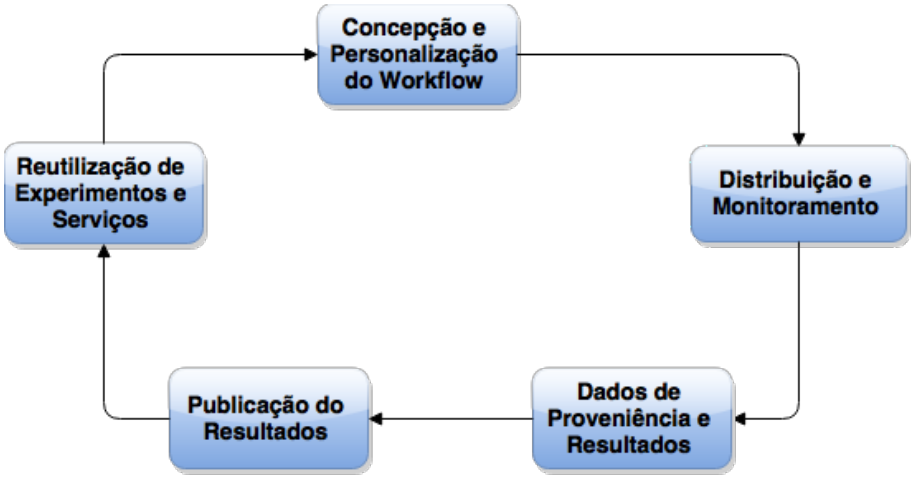


Figura 2.7: Ciclo de um experimento científico (OINN *et al.*, 2007) .

Esse modelo inicia na fase de concepção do *workflow* onde pode-se criar um novo ou adaptar um existente para realizar a função planejada. Essa fase é seguida pela distribuição do *workflow* aos envolvidos e monitoramento da sua execução no contexto do experimento. Após essa etapa, são coletados os resultados do experimento e as informações de proveniência. Esses resultados são então publicados para a comunidade e o *workflow* pode ser reutilizado em outro experimento ou servir como base para a concepção de um novo.

Com base no modelo proposto por Oinn *et al.* (2007), Mattoso *et al.* (2009) apresentam os desafios de um ciclo de experimentação definido em 3 fases principais: composição, execução e análise. Cada fase apresenta suas tarefas específicas, sendo que as tarefas de "Proveniência" e "Gerência de Configuração" acompanham todo o processo de experimentação, como mostra a Figura 2.8.

Já Belloum *et al.* (2011) descrevem o ciclo de experimentação como iniciando na fase de investigação do problema, onde ocorre a definição do escopo de pesquisa, avançando para a fase de prototipação do experimento, onde são desenvolvidos os componentes e os *workflows* necessários para o experimento. Logo após, o experimento é executado de forma controlada com os dados coletados e finaliza com a publicação dos resultados obtidos, sendo que todas as etapas utilizam repositórios compartilhados, conforme Figura 2.9.

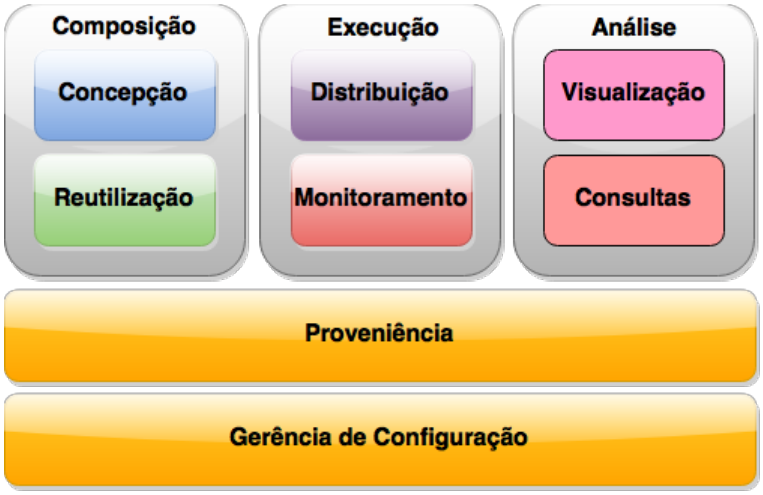


Figura 2.8: Etapas da experimentação (MATTOSO *et al.*, 2009) .

Figura 2.9: Ciclo de vida de um experimento científico (BELLOUM *et al.*, 2011) .

Freitas *et al.* (2015) utilizam os conceitos do ciclo de vida do experimento científico proposto por Belloum *et al.* (2011) para estender este ciclo de vida para o contexto de ecossistemas de software. Nesse ciclo de vida, Freitas *et al.* (2015) propõem a realização de revisões sistemáticas de literatura na etapa de investigação do problema e o uso de conceitos de linha de produto de software científico (LPSC) na fase de prototipação do experimento, conforme Figura 2.10. Devido o trabalho de Freitas *et al.* (2015) se tornar um ecossistema de software científico, seu nome foi alterado de Ecos PL-Science para E-SECO.

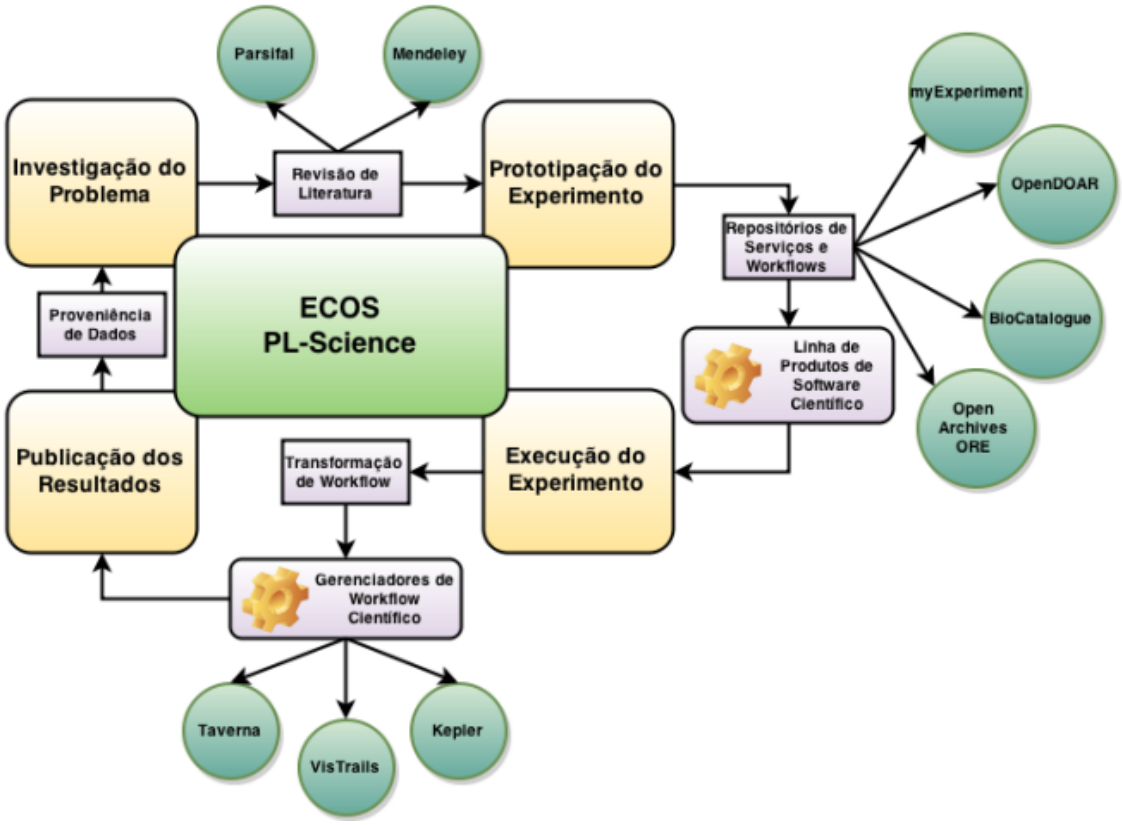


Figura 2.10: Ciclo de vida de um experimento científico no E-SECO (FREITAS *et al.*, 2015) .

Freitas *et al.* (2015) enfatizam a necessidade de colaboração entre os participantes de um ecossistema de software científico e esta preocupação é refletida no ciclo de vida proposto, com ênfase no uso de múltiplos *workflows* ao longo do ciclo de experimentação e as diversas formas de colaboração entre os participantes do experimento. Esta abordagem de Freitas *et al.* (2015), enfatiza ainda mais a necessidade de controle e gerência das diversas versões dos *workflows* associados e a captura dos processos e dados para posterior análise. Por conta disso, uma etapa também importante é a da gerência da proveniência de dados. Detalhamos a seguir esta importante área de pesquisa no contexto de experimentos científicos.

2.2 PROVENIÊNCIA DE DADOS

O termo "proveniência" remete a origem ou procedência, e quando trata-se de proveniência de dados, é um registro da história da derivação dos dados, que possibilita a reprodutibilidade, interpretação dos resultados e diagnóstico de problemas (LIM *et al.*, 2010) .

Assim, proveniência de dados é uma forma para explicar como um produto de dados em particular foi gerado, detalhando as etapas do processo computacional que o produziu, através de informações advindas de metadados.

Alguns SGWfC como o Taverna (OINN *et al.*, 2007) , Kepler (ALTINTAS *et al.*, 2004) e Pegasus (DEELMAN *et al.*, 2004) , permitem capturar os passos do *workflow* durante sua execução. No entanto, esses sistemas muitas vezes adotam modelos proprietários para capturar a proveniência gerada nas execuções (CUEVAS-VICENTTÍN *et al.*, 2014) . O uso de modelos padronizados para a captura de proveniência é uma das alternativas para facilitar a interoperabilidade entre sistemas. O registro da proveniência pode se tornar ainda mais útil se aplicado a laboratórios colaborativos (OLSON, 2009) , onde pesquisadores geograficamente dispersos estão trabalhando em um mesmo experimento, as atividades e dados de análises devem ser registrados e compartilhados com os demais pesquisadores do grupo, evitando a perda de conhecimento e do controle sobre os dados do experimento.

Existem dois tipos de proveniência: retrospectiva e prospectiva. A proveniência retrospectiva pode ser entendida como a captura dos passos que foram executados, bem como as informações sobre os ambientes de execução utilizados. Já a proveniência prospectiva é definida como a captura da especificação de uma tarefa computacional (ou seja, um *workflow*), correspondendo aos passos que precisam ser seguidos para gerar um produto de dados ou classe de produtos de dados (LIM *et al.*, 2010) .

Atualmente, existem dois modelos padrão para a captura de dados de proveniência, o modelo OPM (MOREAU *et al.*, 2011) e o modelo PROV (MOREAU; MISSIER, 2013) . No contexto deste trabalho, o modelo de proveniência de dados utilizado foi o PROV (MOREAU; MISSIER, 2013) , que entende-se como uma evolução do OPM e permite novas relações para representação do conhecimento.

As informações de proveniência de dados podem ser usadas para aprender métodos e regras de design de *workflow* e auxiliar os usuários na criação de *workflows* semelhantes, na compreensão de correlações de dados e na experiência para futuras execuções (MOREAU; FOSTER, 2006) . Considerando este contexto, os *workflows* são adaptados a todo momento dentro de um ciclo de experimentação, e o controle dessas alterações são necessárias para permitir que o estudo seja replicado. Para isso, cada configuração do

experimento e versão do *workflow* deve ser armazenada junto aos seus parâmetros e recursos, necessários para execução. Uma das formas para se armazenar estas informações é através do uso de proveniência.

Para redução e otimização do armazenamento, por conta do grande volume de dados gerados, pode-se capturar apenas os *workflows* estáveis dentro do processo de experimentação, assim seria possível reduzir o volume de dados e otimizar a análise dos *workflows* e do experimento.

2.2.1 PROV

Conforme já dito, na busca por interoperabilidade dos dados de proveniência entre diferentes sistemas, pesquisadores da área promoveram a criação de modelos padronizados de proveniência. Dentre estes, merecem destaque o OPM (Open Provenance Model) (MOREAU *et al.*, 2011) e posteriormente o PROV (MOREAU; MISSIER, 2013) , padronizado pela W3C.

O modelo de proveniência PROV é mais abrangente, comparado ao OPM, e possibilita novas formas de representação do conhecimento. Assim, para o estudo sobre a gerência de experimentos científicos, optamos por utilizar o modelo PROV, pelo fato de ser mais amplo e permitir a captura tanto de proveniência centrada em processo, quanto centrada em entidade ou centrada em agente.

O modelo PROV possui um conjunto de 12 documentos para auxiliar na especificação de modelo de proveniência, conforme Figura 2.11. Entre os principais documentos podem ser citados o PROV-DM, que especifica o modelo de captura de dados, o PROV-CONSTRAINTS, que é um conjunto de restrições aplicáveis ao modelo de dados (PROV-DM) e o PROV-O, uma ontologia owl2 [2] para mapeamento do modelo de dados.

O modelo de dados PROV distingue núcleos estruturais de estruturas estendidas. Os núcleos estruturais constituem a essência das informações de proveniência, e são comumente encontrados em vários vocabulários específicos do domínio que lidam com proveniência ou tipos semelhantes de informação. O PROV-DM possui uma separação entre tipos e relações no modelo, onde os tipos são (MOREAU; MISSIER, 2013) :

[2]OWL é uma linguagem para definir e instanciar ontologias.

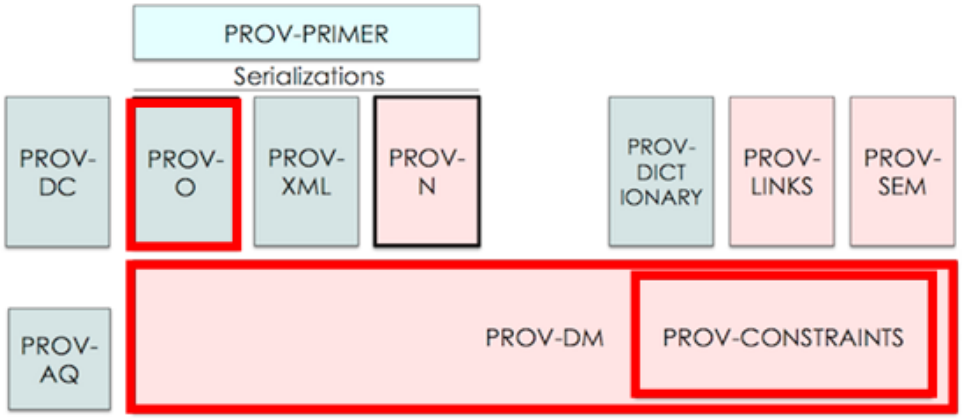


Figura 2.11: Organização do PROV.

- Entidade: é um tipo físico, digital, conceitual ou algo com aspectos fixos, e entidades podem ser reais ou imaginárias;
- Atividade: é algo que ocorre ao longo de um período de tempo e atua sobre entidades; pode incluir consumo, processamento, transformação, modificação, uso ou geração de entidades;
- Agente: é algo que tem algum tipo de responsabilidade por atividade, para a existência de uma entidade, ou para a atividade de outro agente.

Já as Relações existentes no PROV-DM podem ser primárias ou secundárias como representação opcional. As relações primárias são apresentadas na Figura 2.12 e as relações secundárias na Figura 2.13.

Com base nessas relações é composto o núcleo estrutural do PROV, conforme Figura 2.14.

		Object		
		Entity	Activity	Agent
Subject	Entity	**WasDerivedFrom** Revision Quotation PrimarySource AlternateOf SpecializationOf HadMember	**WasGeneratedBy** WasInvalidatedBy *R T L*	**WasAttributedTo**
	Activity	**Used** WasStartedBy WasEndedBy *R T L*	**WasInformedBy**	**WasAssociatedWith** *R*
	Agent	—	—	**ActedOnBehalfOf**

Figura 2.12: Relações primárias do PROV.

		Secondary Object		
		Entity	Activity	Agent
Subject	Entity	—	WasDerivedFrom (activity)	—
	Activity	WasAssociatedWith (plan)	WasStartedBy (starter) WasEndedBy (ender)	—
	Agent	—	ActedOnBehalfOf (activity)	—

Figura 2.13: Relações secundárias (opcionais) do PROV.

O D-PROV (MISSIER *et al.*, 2013) é caracterizado como uma extensão da especificação PROV da W3C, voltada para trabalhar especificamente com *workflows*. Sua principal característica é a expansão das entidades para a captura de proveniência prospectiva e retrospectiva do *workflow*. Para isso, 6 entidades foram adicionadas ao modelo original para representar a proveniência prospectiva englobando tarefa, relação entre as tarefas e os canais de comunicação. Na proveniência retrospectiva foram adicionadas 4 entidades para registrar as portas de comunicação e ocorrência de leitura e escrita pelas tarefas.

O objetivo com esta expansão foi facilitar a separação dos dados de proveniência prospectiva e retrospectiva, pois a proveniência prospectiva representa o processo (*workflow*) e a retrospectiva representa os dados produzidos em uma execução do processo.

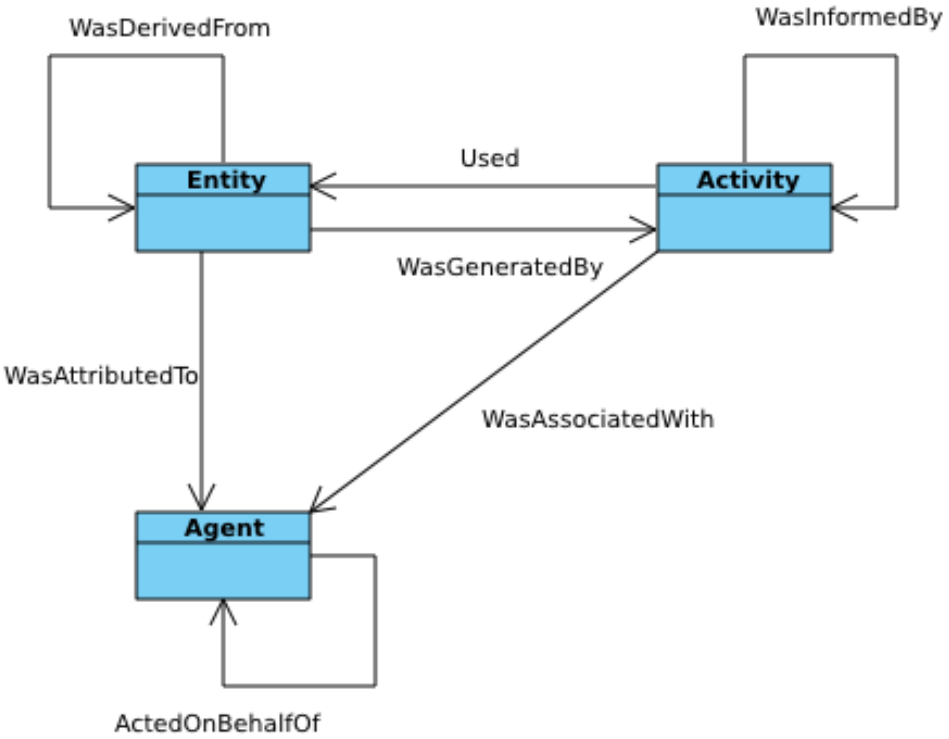


Figura 2.14: Núcleo do PROV.

Apesar de ser uma solução voltada para a proveniência do *workflow*, esta se diferencia da proposta deste trabalho por não registrar informações referentes ao experimento, o que no contexto de experimentação, conforme dito anteriormente, se mostra cada vez mais necessário.

Outro componente importante da especificação do PROV é o documento que trata de uso de ontologias, PROV-O. Uma ontologia, de modo geral, se preocupa com a identificação dos tipos de objetos e como descrevê-los (ANTONIOU; HARMELEN, 2004) . Em computação, uma ontologia é descrita como uma especificação formal e explícita de uma conceituação compartilhada (GRUBER, 1995) . A utilização de ontologias possibilita o compartilhamento de conhecimento sobre os conceitos de um determinado domínio, reutilização do conhecimento e processamento de máquina (YU, 2011) .

A Ontologia PROV (PROV-O) expressa o Modelo de Dados PROV (PROV-DM) usando OWL2 (Web Ontology Language) (LEBO *et al.*, 2013) . Fornece um conjunto de classes, propriedades e restrições que podem ser usadas para representar e trocar informações de procedência gerada em diferentes sistemas e em diferentes contextos, conforme ilustra a Figura 2.15.

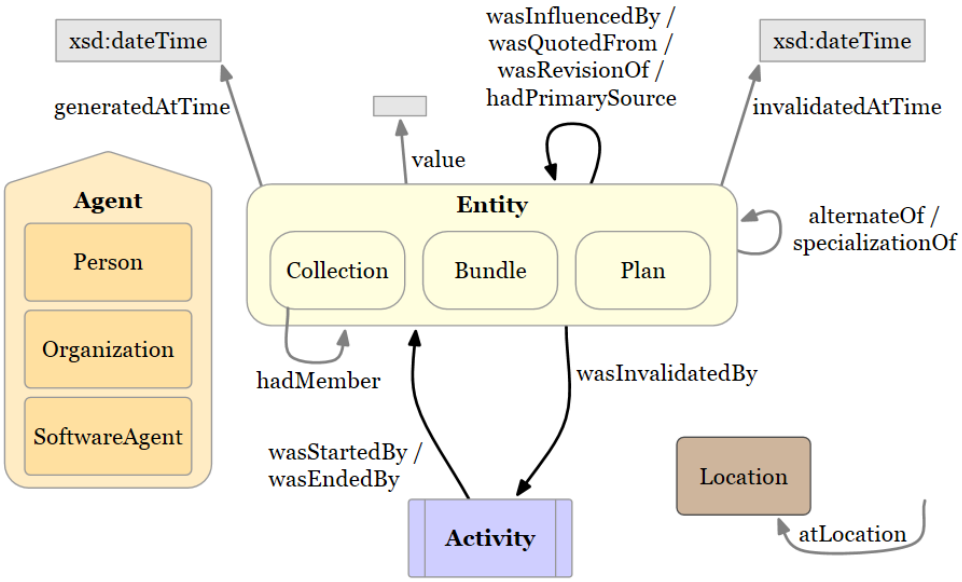


Figura 2.15: Representação do PROV-O.

Apesar de fornecer diversas construções importantes para derivação de conhecimento, a PROV-O não expressa todo o conhecimento necessário para a gerência tanto de experimentos quanto dos *workflows* associados. Uma maneira de trazer esse suporte é através da proposição de regras ontológicas específicas relacionadas a este contexto. Neste sentido, esta dissertação propõe uma extensão da PROV-O que permite expressar estes conhecimento. Esta extensão, denominada PROV-OEXT será apresentada no capítulo 4.4.1.2.

Considerando a importância da captura da proveniência de dados para a gerência de experimentos científicos, foi realizado um levantamento para identificar como os SGWfCs existentes tratam a proveniência dos dados.

Entre os SGWfC analisados estavam o Taverna (OINN *et al.*, 2007) , o Kepler (ALTINTAS *et al.*, 2004) e o VisTrails (FREIRE *et al.*, 2006) . Com esta análise, identificou-se que no Taverna e no Kepler, toda a coleta dos dados de proveniência era realizada por meio de um plug-in instalado a parte. Além disso, no caso do Kepler, não segue nenhum modelo de proveniência de dados conhecido. O Taverna já utiliza o modelo PROV, mas armazena os dados em um Sistema Gerenciador de Banco de Dados (SGBD) na própria ferramenta, permitindo a extração dos dados por meio de arquivos xml ou xls, o que dificulta a troca de informações entre os sistemas e, consequentemente, a análise dos dados pelos pesquisadores. Infelizmente, na prática, não foi possível analisar o uso desses plug-ins pois, ao instalá-los, os SGWfCs deixavam de operar.

2.3 MANUTENÇÃO E EVOLUÇÃO DE SOFTWARE

O processo de desenvolvimento de um *workflow* pode ser comparado ao de um software, onde o desenvolvimento deve ser organizado em partes e a forma de se organizar e relacionar estas partes denominamos paradigma de ciclo de vida. Independente do processo, uma grande quantidade de dados é produzida e pode apresentar diferentes níveis de relevância para o processo. Esses dados selecionados e agrupados definem o que no desenvolvimento de software é chamado de configuração de software (CIA, 2006) .

Neste contexto, a gerência de configuração surgiu da necessidade de controle das alterações, através do uso de métodos e ferramentas, buscando-se maximizar a produtividade e minimizar os erros cometidos, em consequência das manutenções e evoluções que ocorrem ao longo do ciclo de vida. A atividade de manutenção de software é caracterizada pela modificação de um produto de software já entregue ao cliente para correção de erros, melhora no desempenho ou adaptação do produto a um novo ambiente (MAMONE, 1994) . O impacto dessas mudanças em um software podem ser definidas como atividades de evolução de software, conforme Sommerville (2011)).

Lehman (1996) publicou estudos apresentando as "Leis de Evolução de Software", nas quais são abordados 8 fatores que normalmente ocorrem em manutenção de software. Não é objetivo desse trabalho discutir o termo "Leis", mas sim utilizar-se do arcabouço conceitual apresentado por esses estudos. Alguns destes fatores podem ser abordados no contexto de manutenção e evolução de *workflows*:

- A "Lei da Mudança Contínua" afirma que todo software tem que ser modificado continuamente ou se tornará menos útil. Aplicando ao contexto de *workflows*, à medida que as pesquisas avançam, os *workflows* utilizados tendem a necessitar de evolução, para manterem-se úteis ao experimento;
- A "Lei do Incremento de Complexidade" define que, com o processo de evolução, a estrutura tende a se tornar mais complexa e necessitar de maior atenção;
- Durante o ciclo de vida de um software, mudanças incrementais são constantes, ou seja, a "Lei de Crescimento Contínuo", no contexto de *workflow*, o mesmo deve evoluir conforme o experimento, apresentando novas funcionalidades.

De modo geral, a gerência de configuração visa controlar tanto a manutenção como a evolução, evitando a inconsistência dos dados e focando no controle das mudanças que ocorrem durante o processo (WHITGIFT, 1991) . Conforme Estublier (2000) a gerência de configuração de software busca prover serviços para apoiar este processo.

Alguns dos conceitos utilizados na gerência de configuração de software de forma geral podem ser aplicados e utilizados no contexto de experimentação científica e utilizados na proposta do E-SECO ProVersion. São eles: i) Gerenciar repositórios; ii) Apoiar as atividades usuais; iii) Apoiar e controlar o processo.

Conforme Dantas (2009), a utilização da gerência de configuração no controle das manutenções e evoluções podem apresentar vantagens como:

- Ganho de produtividade: visto que se conhece todos os passos executados e evita-se retrabalho;
- Diminuição do retrabalho e erros: a inserção de erros e alto comum de ocorrer durante processo de manutenção ou evolução, com o reuso de componentes reduz o trabalho e evita-se a criação de erros;
- Aumento da disciplina no processo: uma vez que o processo é monitorado para verificar se segue conforme previsto, todas as atividades passam a seguirem um modelo para suas realizações, o que torna o processo mais disciplinado;
- Aumento da memória organizacional: Com o avanço dos laboratórios colaborativos, a gestão do conhecimento entre todos os membros se torna importante para o controle do andamento do processo;
- Acesso as informações referentes ao processo: para que todos tenham conhecimento sobre o andamento do processo, todos devem ter acesso a memória de grupo e verificar quais passos foram executados e quais estão para serem executados;
- Possibilita estabelecer uma trilha para auditoria: o uso de laboratórios colaborativos, pode causar um descontrole do processo e para que todos os dados possam ser validados é necessário que os passos possam ser reproduzidos, onde para isso é importante o registro dos mesmos;

- Auxilia a gerência de projetos: a gerência de configuração é uma etapa que contribui para o projeto como um todo, através do registro de mudanças, acompanhamento das manutenção e evoluções existentes;

- Garante um ambiente estável: a gerência de configuração deve registrar todos os acontecimentos dentro do processo e com isso garantir que o mesmo seja executado conforme foi planejado.

Todas essas características de processos de software podem ser aplicadas para apoiar a gerência de configuração dos experimentos, junto com a manutenção e evolução dos *workflows* integrantes do mesmo, dando apoio a proposta do E-SECO ProVersion objeto deste trabalho.

2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou os pressupostos teóricos deste trabalho, envolvendo os ciclos de vida de experimentos e de *workflows* científicos. Também foram apresentados alguns SGWfC e discutido como os mesmos tratam do controle dos dados gerados pelos *workflows* durante a execução do experimento. Foram apresentados ainda os principais modelos de proveniência dando maior foco no PROV, que foi o modelo empregado na proposta deste trabalho, explicando suas regras e funcionalidades. Ao final foram discutidos alguns conceitos sobre manutenção e evolução de software, e de gerência de configuração, que contribuem para a proposta do E-SECO ProVersion.

No capítulo seguinte são apresentados os trabalhos relacionados e uma comparação dos mesmos.

3 TRABALHOS RELACIONADOS

Neste capítulo é apresentada uma revisão *quasi* sistemática relacionada ao tema, o que permitiu identificar a relevância da pesquisa e os principais trabalhos relacionados.

3.1 REVISÃO *QUASI* SISTEMÁTICA

No contexto desta dissertação, é importante identificar na literatura como os pesquisadores gerenciam o ciclo de vida dos experimentos científicos e dos *workflows* vinculados às pesquisas, de forma a identificar pontos não cobertos e/ou falhos. Para isso foi realizada uma revisão *quasi* sistemática da literatura.

Segundo (TRAVASSOS *et al.*, 2008) , uma revisão *quasi* sistemática é um estudo secundário que tem como objetivo identificar, avaliar e interpretar trabalhos existentes na literatura. A diferença entre a revisão sistemática tradicional com a *quasi* revisão sistemática, é a caracterização principal do estudo, onde a *quasi* revisão é a caracterização de uma área, ou a não existência de objetos para comparação, conforme Kitchenham *et al.* (2009).

Considerando o objeto de estudo desta dissertação, esta revisão *quasi* sistemática buscou identificar quais os principais SGWfC existentes, como os mesmos tratam as mudanças que um *workflow* sofre durante uma experimentação, como são armazenados e distribuídos os dados do experimento entre a comunidade científica e a colaboração entre os pesquisadores do mesmo domínio. Esse tipo de informação se faz importante pois, para que outros pesquisadores possam reutilizar um serviço, *workflow* ou resultados de um experimento, devem conhecer os mesmos.

A Engenharia de Software (ES) vem utilizando a experimentação como instrumento para a criação de um corpo de conhecimento e, para que apresente validade científica, todos seus itens precisam ser verificados perante a realidade através de estudos experimentais (JURISTO; MORENO, 2013) , algo comum nas pesquisas e no uso da revisão da literatura.

Uma revisão sistemática da literatura é um dos meios existentes para identificar, avaliar e interpretar toda pesquisa pertinente a uma pergunta de pesquisa em particular (KITCHENHAM *et al.*, 2009) . Outras razões mais específicas que justificam o uso da revisão sistemática de acordo com Kitchenham *et al.* (2009) são:

- Resumir alguma evidência existente sobre uma determinada teoria ou tecnologia;
- Identificar uma ramificação em aberto, de uma determinada linha de pesquisa em questão, possibilitando a definição de áreas onde mais investigações podem ser exploradas.

O escopo para aplicação dessa revisão *quasi* sistemática relaciona-se a métodos utilizados em processos de *workflows* científicos, focando na identificação de métodos, ferramentas, modelos e processos, utilizados em e-Science atualmente, com o objetivo de caracterizar o estado da arte.

Steinmacher, Chaves e Gerosa (2013) apresentam alguns pontos importantes para uma revisão sistemática e que foram ajustados para esta pesquisa. As questões formuladas a seguir servem de base de conhecimento sobre o tema pesquisado, por responderem perguntas que vão auxiliar a pesquisa como um todo.

As questões a serem respondidas pela revisão *quasi* sistemática são:

- Questão 1: Como as publicações sobre processo de *workflow* científico vem evoluindo ao longo dos anos? Espera-se que a resposta a essa questão aponte o crescimento ou encurtamento da área de pesquisa.
- Questão 2: Quem são os autores mais ativos na área? Com essa pergunta, espera-se ter uma indicação dos principais pesquisadores da área, oferecendo uma referência para as publicações relacionadas em pesquisas futuras.
- Questão 3: Quais Conferências ou Congressos são os principais alvos para a pesquisa e publicação na área? Ao responder a essa pergunta, espera-se um norte de onde possam ser encontrados mais artigos sobre o tema, bem como identificar bons alvos para publicação.

- Questão 4: Quais linhas de pesquisa dentro da área de e-Science, estão tendo maior interesse pela comunidade científica internacional? Espera-se com essa resposta, identificar estudos que possam ser relevantes para a área, evitando estudos que pouco venham a contribuir com a comunidade científica de forma geral.

Essas perguntas servem para caracterizar a área de pesquisa. Outras perguntas com um enfoque mais preciso a esta pesquisa, são:

- Questão 5: Como vem sendo tratada a gerência do ciclo de vida de um *workflow* dentro do ciclo de experimentação? Espera-se com essa resposta identificar algum método, ferramenta, framework ou modelo que classifique as mudanças que os *workflows* sofrem ao longo do seu ciclo de vida.

- Questão 6: Como são armazenados os *workflows* científicos, bem como os dados do experimento ao longo do seu ciclo de experimentação? Com essa resposta espera-se entender como são armazenados os dados do experimento, as ferramentas usadas e como são compartilhadas as informações com a comunidade científica.

3.1.1 ESTUDOS PRELIMINARES

O estudo preliminar contou com a seleção das fontes de pesquisa, onde foram selecionadas somente as bases de dados cujo conteúdo encontra-se disponível através do acesso de internet disponibilizado pela UFJF (Universidade Federal de Juiz de Fora). Foram selecionadas 6 bases de dados apresentadas por Brereton *et al.* (2007) e Steinmacher, Chaves e Gerosa (2013), que atende ao critério de acesso pela UFJF, e ainda os anais de conferências nacionais (BDBComp) (LAENDER; GONÇALVES; ROBERTO, 2004) . As bases de dados utilizadas foram: (i) IEEE Digital Library; (ii) ACM Digital Library; (iii) Scopus; (iv) CiteSeerX; (v) ISI Web Of Science; (vi) Google Scholar; e os anais de conferências nacionais (vii) BDBComp;

Ficou definido que os idiomas de estudo seriam o Inglês e o Português. O inglês por ser considerado como linguagem padronizada internacionalmente e o Português por conta de algumas bases utilizadas no estudo permitirem a indexação de artigos nessa língua, além de ser a língua oficial do país do estudo.

As buscas nas bases foram feitas utilizando a String de busca criada com o método PICOC. O PICOC é um acrônimo para "Population, Intervention, Comparison, Outcome e Context" sendo um método usado para descrever uma pergunta pesquisável (SILVA *et al.*, 2010) . A String de busca é formada pelas palavras-chave apresentadas na Tabela 3.2 e executada sobre as bases listadas anteriormente. Os artigos recuperados foram catalogados utilizando o Parsifal[1]. Para as bases de dados da IEEE Digital Library, CiteSeerX, e Google Scholar, Scopus, ACM Digital Library e ISI Web Of Science, as buscas foram realizadas utilizando o modo de busca avançado, disponível em cada um dos portais, sendo que a String foi adaptada para as particularidades de cada uma das bases. No caso do BDBComp, como a base não aceita a String utilizada nas demais bases, a busca foi realizada por partes, utilizando as palavras-chave que compõem a String.

Para o mapeamento da pesquisa, foram aplicadas, com base nos critérios definidos durante a preparação do escopo, questões que deveriam ser respondidas. Cada um dos critérios de mapeamento foi utilizado para a construção da String, conforme o método PICOC e tinham como objetivo responder as intenções a seguir:

1. Identificar como são armazenados os dados do experimento e como foram gerenciadas as manutenções e evoluções durante o ciclo de experimentação, bem como o motivo destas manutenções e evoluções, além do que as motivou;

2. Identificar métodos, ferramentas, técnicas, modelos ou frameworks para o gerenciamento dos *workflows* e dos experimentos científicos.

3.1.2 CRITÉRIOS DE REFINAMENTO DOS ESTUDOS

Os artigos foram analisados e selecionados baseados inicialmente pelo critério de análise das palavras-chave, título do artigo e *abstract*. A segunda análise foi fundamentada no contexto abordado pelo artigo, o meio de publicação, os autores e, por fim, o tipo de artigo, que pode ser somente teórico ou teórico com prática.

Esses refinamentos foram feitos em 3 etapas, onde buscou-se, após o processo de eliminação dos artigos duplicados e da seleção por palavras-chave, título e abstract, ponderá-los em notas que variavam em uma escala de 0 à 2, com intervalos de 0,5, utilizando-se do

[1]Ferramenta online de gerenciamento de revisões bibliográficas (http://parsif.al/).

recurso de classificação do Parsifal. Assim, classificou-se como úteis à pesquisa apenas os que trabalhos que obtiveram uma nota igual ou superior a 1.

Os resultados da busca foram gerenciados utilizando a ferramenta Parsifal e analisados para a extração do conteúdo considerado útil para o estudo do gerenciamento de *workflows*, com base nos critérios apresentados. Essa sumarização, foi baseada na definição e execução da String, onde para a criação da mesma, como já dito, foi aplicado o método PICOC para definição e montagem da mesma. Para auxiliar nessa função, utilizou-se o método GQM (Goal, Question, Metric) (BASILI, 1993) .

A Tabela 3.1 define o objetivo específico do estudo, baseado na abordagem GQM conforme Wohlin *et al.* (2012). O GQM é um enfoque da mensuração orientada em metas que ajuda na definição e implementação de uma questão de pesquisa (WANGENHEIM; RUHE, 1999) .

Tabela 3.1: Objetivos específicos de estudo

Analisar:	Abordagens, métodos e ferramentas, utilizadas para gerenciar experimentos
Com o propósito de:	Caracterizar
Em respeito a:	Evolução e manutenção dos experimentos de e-Science
Do ponto de vista dos:	Pesquisadores
No contexto do:	Ciclo de experimentação científica

Para a construção do PICOC, utilizou-se uma busca abrangente com o intuito de apresentar o estado da arte. A Tabela 3.2 apresenta o PICOC com as respectivas palavras-chave em inglês utilizadas para a construção da String de busca. Todas as palavras-chave foram definidas em inglês, pelo fato dos artigos escritos em português possuírem título e/ou abstract em inglês.

Tabela 3.2: Resultado da separação das palavras-chave no PICOC

PICOC:	Palavras-Chave
P (População)	scientific workflow, scientific experiment, science process, scientific process, workflow in e-science, e-science workflow
I (Intervenção)	development, modeling, specification, design, definition, conception, description, analysis, representation, approach, method, technique, model, process, tool
C (Comparação)	Não se aplica
O (Saída)	approach, method, technique, model, process, framework, tool, support
C (Contexto)	scientific workflow, scientific experiment, science process, scientific process, workflow in e-science, e-science workflow

Com base nas palavras-chave definidas no PICOC, foram agrupadas as palavras-chave da população, mesclando a intervenção com a saída. Assim criou-se a String de busca conforma apresentada a seguir:

"*((scientific workflow) OR (scientific experiment) OR (science process) OR (scientific process) OR (workflow in e-science) OR (e-science workflow)) AND ((development approach) OR (development method) OR (development technique) OR (development model) OR (development process) OR (composition approach) OR (composition method) OR (composition technique) OR (composition model) OR (composition process) OR (modeling approach) OR (modeling method) OR (modeling technique) OR (modeling model) OR (modeling process) OR (specification approach) OR (specification method) OR (specification technique) OR (specification model) OR (specification process) OR (design approach) OR (design method) OR (design technique) OR (design model) OR (design process) OR (definition approach) OR (definition method) OR (definition technique) OR (definition model) OR (definition process) OR (conception approach) OR (conception method) OR (conception technique) OR (conception model) OR (conception process) OR (description approach) OR (description method) OR (description technique) OR (description model) OR (description process) OR (analysis approach) OR (analysis method) OR (analysis technique) OR (analysis model) OR (analysis process) OR (representation approach) OR (representation method) OR (representation technique) OR (representation model) OR (representation process))*"

Para a concepção dessa string de busca, utilizou-se um conjunto com três artigos de

controle, que deveriam ser retornados na busca no momento de execução nas bases. Esses artigos estão apresentados na Tabela 3.3.

O primeiro artigo, "Designing the myExperiment Virtual Research Environment for the Social Sharing of Workflows" (ROURE; GOBLE; STEVENS, 2009) , foi escolhido por abordar sobre o armazenamento e o apoio ao compartilhamento de *workflows* científicos através do uso do myExperiment (GOBLE; ROURE, 2007) . O segundo artigo, "Towards supporting the life cycle of large scale scientific experiments" (MATTOSO *et al.*, 2009) , foi escolhido por abordar o ciclo de vida dos experimentos científicos, apresentando algumas das dificuldades que existem principalmente em relação à proveniência dos dados e com o gerenciamento dos *workflows*. Por fim, o terceiro artigo, "Mining usage patterns from a repository of scientific workflows" (DIAMANTINI; POTENA; STORTI, 2012) , trata do uso de repositório de *workflows* científicos e como o mesmo pode ser utilizado para apoiar pesquisa em e-Science.

Tabela 3.3: Artigos de controle

Artigo	Ano	Local de Publicação
Designing the myExperiment Virtual Research Environment for the Social Sharing of Workflows e-Science and Grid Computing	2007	IEEE International Conference on
Towards supporting the life cycle of large scale scientific experiments Process Integration and Management	2010	International Journal of Business
Mining usage patterns from a repository of scientific workflows on Applied Computing	2012	27th Annual ACM Symposium

3.1.3 RESULTADOS DA BUSCA

Para cada base pesquisada, foram feitos alguns levantamentos, onde buscou-se responder as questões apresentadas na seção 3.1. Os dados sobre a evolução das publicações são apresentados em um gráfico onde se pode analisar a sua evolução ao longo dos anos.

O resultado dessa análise está apresentado na Figura 3.1. Com base na Figura 3.1, é possível perceber o crescimento da área nos últimos 15 anos, consolidando a importância

de estudo envolvendo o tema em questão.

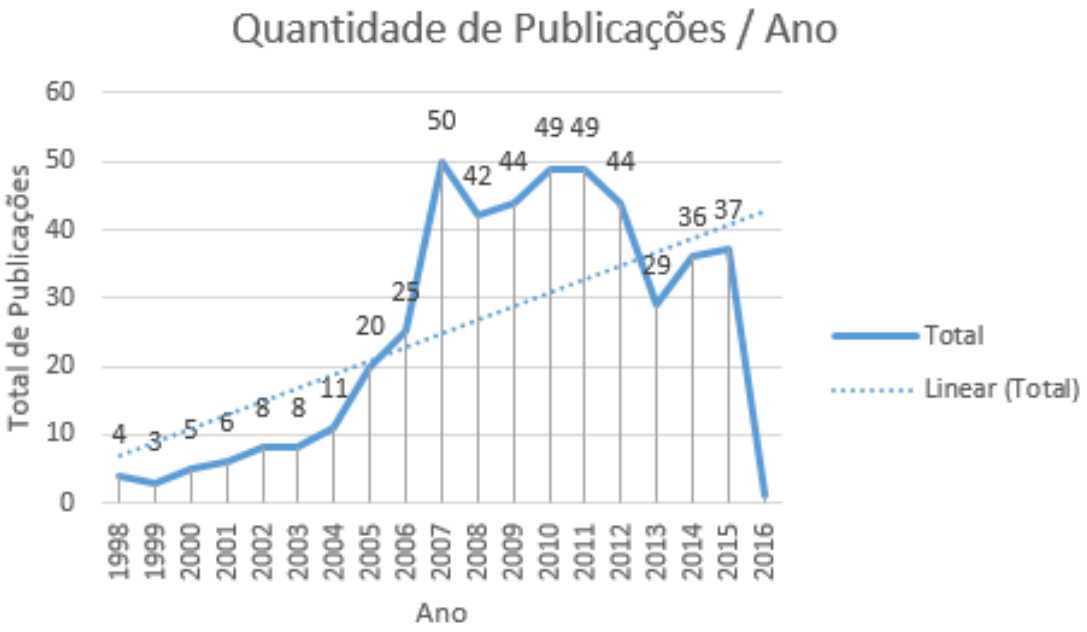


Figura 3.1: Evolução das publicações ao longo dos anos.

Conforme apresenta a Figura 3.1, pode-se analisar que, a partir do ano de 2004, o número de publicações começou a crescer. Isso pode ser caracterizado pela criação do *IEEE International Conference on e-Science* em 2004 e em 2007 pela criação do *e-Science Workshop*, conforme identificado na pesquisa.

A revisão sistemática foi realizada em duas datas distintas, a primeira em 22/07/2014 e posteriormente em 05/05/2015 para atualização da revisão. Não se pode afirmar que houve uma queda das pesquisas da área, visto que não se tem dados de que todas as publicações desses últimos anos foram efetivamente indexadas pelas bases de busca, como o caso observado do BDBComp, onde no momento da segunda busca não foi identificado nenhum artigo novo.

Após ter a catalogação de cada um dos resultados lançados no Parsifal, foram aplicados os filtros nos resultados a fim de eliminar resultados repetidos, fora do escopo da pesquisa e que não se enquadravam nos critérios de seleção definidos.

O primeiro filtro considera a eliminação dos resultados duplicados que por algum motivo não foram identificados durante o processo de importação do BibTex[2] no Parsifal. Como resultado da busca tinha-se um total de 633 artigos onde, desse total, 19 se en-

[2]É um formato de descrição de bibliografias.

contravam duplicados. O segundo filtro baseou-se na leitura dos títulos e abstracts dos artigos aplicando os critérios definidos. O objetivo desse filtro foi obter apenas os artigos classificados como úteis dentro do contexto da pesquisa por abordarem o tema em questão. Como resultado da aplicação do segundo filtro, o número final de artigos ficou em 76, sendo que ainda ocorreria a classificação dos artigos, cujo objetivo era ponderar apenas os artigos que tratassem do ciclo de vida do *workflow* e sobre repositório de *workflows*. Essa última classificação consistiu em atribuir pesos aos artigos. Cada um dos objetos receberam notas variando de 0 quanto "Não aborda", "Aborda Parcialmente", cujo nota é 0,5 e "Aborda especificamente", cujo nota é 1. Este critério de atribuição de notas é uma forma de classificação do Parsifal, e a nota máxima podia chegar a 2 pontos, por serem 2 questões de qualificação. O terceiro filtro considerou apenas os artigos cuja a classificação tenha sido igual ou superior 1, como já mencionado, baseado em sua leitura. Como resultado desse filtro, restou um total de 17 artigos cujo o conteúdo é relevante para a pesquisa de modo geral.

O terceiro filtro, que foi ponderar os artigos, classificou os artigos que atenderam completamente aos objetivos da pesquisa, os que atendiam parcialmente ou os pouco relevantes para o objetivo do estudo. Assim, os artigos poderiam possuir notas 1, 1.5 ou 2. O resultados dessa análise está disposta na Tabela 3.4.

Tabela 3.4: Artigos ponderados

Ponderação	Total
2	2
1,5	5
1	10

Ao final da análise observou que somente dois dos artigos atendiam completamente o escopo da pesquisa. Desses, dois já eram utilizados como de controle. O primeiro que tratava do compartilhamento de *workflows* científicos e de laboratórios virtuais e o terceiro que tratava extração de dados em repositórios de *workflows* científicos. O segundo artigo utilizado no controle foi classificado como "Parcial", pois o mesmo não aborda o controle de dados do experimento, entretanto trata do ciclo de vida dos experimentos científicos e mais especificamente da gerência de configuração do experimento como algo contínuo no ciclo de experimentação. Com esses 3 artigos foi possível abordar os temas que a proposta E-SECO ProVersion visa atender.

Outra análise realizada foi com relação aos artigos que encontravam-se duplicados entre as bases de busca. Foi realizada uma análise utilizando o diagrama de Venn (VENN, 1880) com o intuito de mapear a fonte desses artigos, como pode ser visto na Figura 3.2.

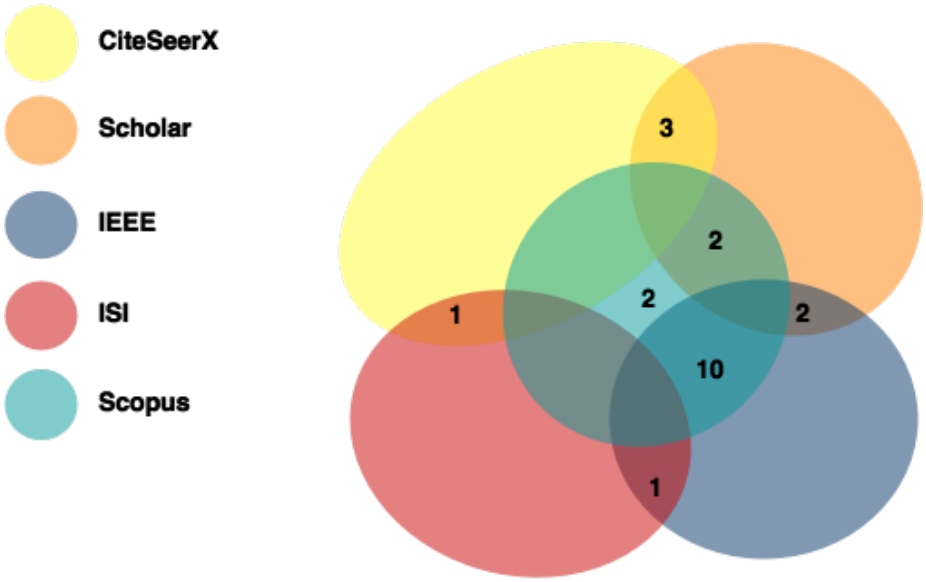


Figura 3.2: Diagrama de Venn dos artigos duplicados.

Com base no diagrama de Venn é possível analisar a repetição dos artigos por base, inclusive dentro da própria base, como no caso da Scopus que apresenta 2 artigos duplicados. Outra base com grande número de repetições é a do IEEE, que teve artigos duplicados entre as base ISI, Scholar e principalmente Scopus. Isso justifica a utilização do primeiro filtro, a fim de eliminar os artigos duplicados. Foi analisado ainda o número de artigos recuperados por base com relação ao número de aceitos de cada uma. O resultado pode ser visto na Figura 3.3.

Como pode ser visto na Figura 3.3, grande parte dos artigos recuperados na busca não atenderam aos requisitos especificados e foram eliminados na aplicação dos filtros de refinamento. Essa análise é baseada na execução do 2^{o} filtro, antes da ponderação dos artigos por significância para a pesquisa. Com base nesses resultados, foram identificadas as conferências e periódicos com maior número de publicações relevantes para o contexto de pesquisa. Com isso, é possível responder a questão 3 levantadas na seção 3.1. O resultado dessa análise é apresentado na Tabela 3.5.

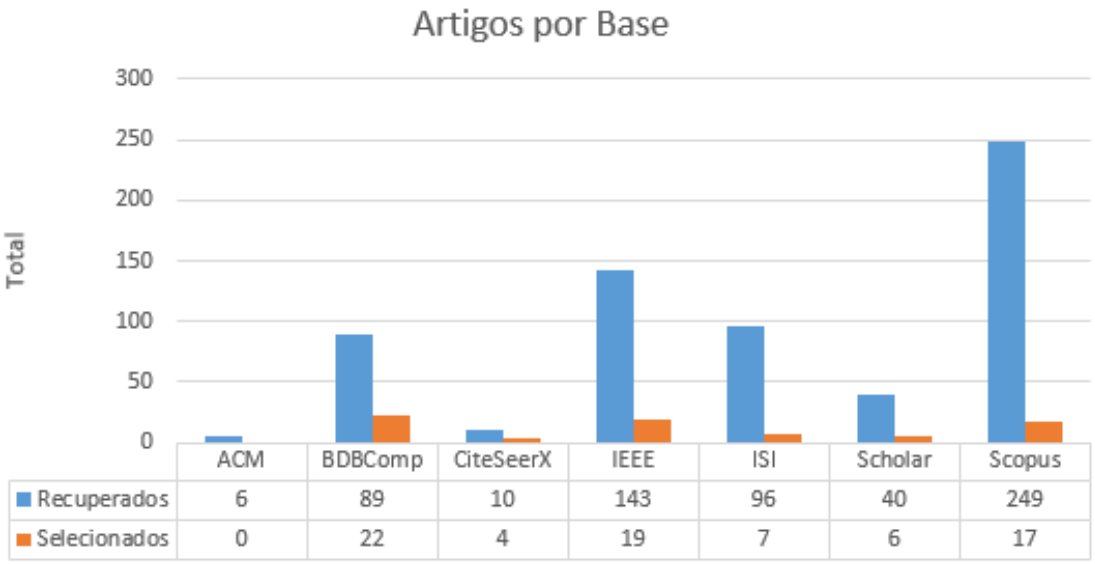


Figura 3.3: Análise dos artigos por base.

Tabela 3.5: Principais meios de publicação

Nome	Tipo	Qualis
Future Generation Computer Systems	Revista Internacional	A2
IEEE International Conference on e-Science and Grid Computing	Conferência Internacional	B1
International Journal of Business Process Integration and Management	Revista Internacional	B4
e-Science Workshop	Workshop Brasileiro	B5
Brazilian symposium on Databases	Simpósio Brasileiro	B2
Experimental Software Engineering Latin American Workshop	Workshop América Latina	B4
Brazilian Symposium on Software Engineering	Simpósio Brasileiro	B2
Workshop on Workflows in Support of Large-Scale Science	Workshop Internacional	B4
IEEE World Congress on Services	Conferência Internacional	B2

Essas informações são úteis para caracterizar possíveis locais de publicações. Junto com essa análise, foi feita a identificação dos principais pesquisadores dentro da área de pesquisa. Os resultados dessa análise são apresentados na Tabela 3.6, com os autores com mais de 10 artigos publicados que foram retornados na busca.

Tabela 3.6: Autores Mais Ativos da Área

Nome	Artigos	H-Index	Local de Trabalho
Deelman, E	29	43	University of Southern California
Goble, C.	22	42	University of Manchester
De Roure, D.	19	31	University of Oxford
Simmhan, Y	19	21	Indian Institute of Science
Mattoso, M	23	20	Universidade Federal do Rio de Janeiro
Ogasawara, E.	12	16	CEFET/RJ
Zhao, Z	29	7	University of Amsterdam

Um detalhe importante que deve ser observado, é que não existe uma relação entre o número de publicações dos autores com o seu H-index. Alguns autores, mesmo com um número menor de publicações, são mais citados entre os pesquisadores. Buscando responder à quarta questão de pesquisa, foram listados quais os temas mais abordados dentro dos artigos. Caracterizando os temas com maior interesse atualmente, os resultados estão dispostos na Tabela 3.7.

Tabela 3.7: Temas Com Maior Publicação

Sistemas de Gerenciamento de *Workflows*
Simulação ou Experimentação através de *Workflows* Científicos
Proveniência de Dados em *Workflows* Científicos
Workflow Científico em Bioinformática

Como resultado dessa análise, foi identificado que poucas publicações tratavam do gerenciamento dos experimentos científico envolvendo os dados do experimentos com os *workflows* vinculados o mesmo, bem como o uso de repositórios. Isso serve de base para as pesquisas futuras abordando essa linha de pesquisa.

3.2 TRABALHOS RELACIONADOS

Com base nos resultados da revisão *quasi* sistemática, alguns trabalhos relacionados foram analisados, considerando a abordagem E-SECO ProVersion. Estes trabalhos foram divididos em duas partes, sendo respectivamente:

i Trabalhos sobre *workflows* que, em geral, tratam o tema de maneira simplificada. Esses trabalhos têm como foco o versionamento, repositórios, armazenamento e o

tratamento da proveniência de dados;

ii Trabalhos que tratam da gestão, manutenção e evolução de software de maneira geral, cujos resultados possam contribuir para a manutenção e evolução de experimentos e *workflows* científicos.

Considerando os trabalhos relacionados a *workflows*, tem-se o trabalho de Santos *et al.* (2008), o qual aborda grafos de *workflow*, e propõe o versionamento de *workflows* utilizando a medida de distância do cosseno. A partir dessa medida, define-se a similaridade entre dois elementos. Jung e Bae (2006), em seu trabalho sobre clusters de *workflows*, focam o apoio à análise de repositórios de processos onde os *workflows* são agrupados, de acordo com o cálculo de similaridade entre as atividades. A semelhança é identificada por um algoritmo de agrupamento hierárquico.

Já Schmidt e Gloetzner (2008) trazem uma abordagem independente do modelo para comparação através do SiDiff. O SiDiff baseia-se principalmente na noção de semelhança entre os elementos do modelo, utilizando algoritmos de comparação e merge na análise dos dados para definição de versionamento.

O myExperiment é um ambiente colaborativo, de compartilhamento e publicação de *workflows* (ROURE; GOBLE; STEVENS, 2009) (GOBLE *et al.*, 2010) , que propõe a separação dos *workflows* científicos, através da organização dos mesmos em grupos, com o intuito de serem reutilizados em outros experimentos (GODERIS *et al.*, 2008) . Permite ainda a interação entre os pesquisadores através da troca de mensagens pessoais. Contudo, não permite que o pesquisador analise os resultados de um experimento, faça *download* de resultados ou acesse repositórios de proveniência diretamente, limitando o conhecimento de pesquisa (MIRANDA *et al.*, 2014) .

O CrowLabs é um repositório similar ao myExperiment, que foi desenvolvido pelo grupo do SGWfC VisTrails (CALLAHAN *et al.*, 2006) . Permite a execução de *workflows*, importação de dados de entrada, análise e reutilização dos dados por terceiros. Sendo os dados disponibilizados aos pesquisadores no formato XML (SANTOS *et al.*, 2009) (MATES *et al.*, 2011) .

O SimiFlow é uma arquitetura para comparação e agrupamento de *workflows* pré-existentes por similaridade, visando a construção de vários experimentos por meio de

abordagem ascendente. Utiliza um algoritmo definido por Seo *et al.* (2007), onde a semelhança é analisada a partir do nome, das portas de entrada e saída existentes nas atividades e do tipo de relacionamento existente entre duas atividades (SILVA *et al.*, 2010) .

O CollabCumulus é um portal que permite ao pesquisador ter acesso ao conteúdo de repositórios de proveniência de forma a realizar a análise dos dados gerados ou consumidos. Cada pesquisador tem um usuário associado que tem acesso a um conjunto de repositórios de proveniência, e pode comentar, criar tópicos de discussão e analisar colaborativamente os resultados com seus parceiros de pesquisa. Além disso, provê busca de informações relativas aos *workflows*, atividades, ativações e arquivos que já foram executados (MIRANDA *et al.*, 2014) .

O Camera II (ALTINTAS *et al.*, 2010) é uma arquitetura para a gerência de *workflows* e componentes do ciclo de experimentação que permite acompanhar os dados do experimento e mapear os dados semanticamente. Apesar de focar na proveniência dos dados, o mesmo não utiliza uma modelo de proveniência, o que restringe a interoperabilidade com diferentes sistemas, e o uso de máquinas de inferência para auxiliar na extração do conhecimento.

O P-GRADE (KACSUK, 2011) , um portal de gerência de experimentos científicos que utiliza grades computacionais com foco na performance de execução dos experimentos científicos. O P-GRADE não foca a proveniência dos dados gerados na execução, apenas em obter um alto poder de processamento para a execução dos *workflows* científicos. Já o gUSE (KAIL *et al.*, 2014) é uma infraestrutura complementar ao P-GRADE criada especificamente para tratar de falhas de execução e resultados incertos de execução.

O PBASE (CUEVAS-VICENTTÍN *et al.*, 2014) é um repositório de *workflows* científicos que usa como base o ProvONE (GROUP *et al.*, 2014) , que é uma extensão do modelo de proveniência PROV destinada a *workflows* científicos, permitindo análise e replicação de experimentos.

O Karma (SIMMHAN *et al.*, 2006) é um framework para captura de proveniência de experimentos científicos focados em *workflows*, utiliza um serviço web para captura dos dados e os armazena em um repositório no formato XML.

O PASOA (Provenance-Aware Service-Oriented Architecture) (GROTH *et al.*, 2006) é um mecanismo de captura de proveniência para experimentos científicos independente do SGWfC. Ele foi desenvolvido para o contexto de bioinformática e todos os dados coletados são armazenados em meta-dados no formato XML.

O myGrid (STEVENS; ROBINSON; GOBLE, 2003) provê um conjunto de ferramentas para dar suporte a construção, gerenciamento e colaboração de experimentos de biologia em ambientes in silico. Utiliza ontologias para permitir o mapeamento dos dados, para que sejam fácilmente descobertos, integrados e compartilhados entre os cientistas (ZHAO *et al.*, 2004) . Entretanto, o mesmo utiliza um modelo próprio para armazenamento dos dados de proveniência, o que dificulta a integração com outras ferramentas.

O D-OPM (CUEVAS-VICENTTIN *et al.*, 2012) é um modelo que estende o OPM, com aspectos específicos de *workflow*, apresentando uma abordagem próxima a deste trabalho, entretanto a proposta E-SECO ProVersion utiliza o modelo PROV, que conforme já dito possibilita um conjunto maior de regras de classificação, facilitando a busca de informações.

O D-PROV (MISSIER *et al.*, 2013) similar ao D-OPM, é um modelo que estende o PROV (MOREAU; MISSIER, 2013) e é voltado para aplicações de *workflow* científico, entretanto, o mesmo se preocupa apenas com os dados de proveniência do *workflow* de forma isolada do experimento ao qual faz parte.

O SciProvMiner (GASPAR *et al.*, 2015) é uma abordagem similar ao D-OPM que estende as regras do OPM para trabalhar com a proveniência de dados de *workflows* científicos. Entretanto, seu foca está na consulta e inferências possibilitadas nos dados coletados e não se preocupa em analisar a evolução e manutenção dos *workflows* nem sua influência sobre a configuração do experimento científico.

O Redux (BARGA; DIGIAMPIETRI, 2008) é uma proposta de representação em camadas para a proveniência em *workflow*, utiliza um modelo abstrato para representação do *workflows* e um modelo próprio para armazenamento dos dados de proveniência.

Marinho *et al.* (2012) apresentam o ProvManager, um sistema de proveniência independente de SGWfC e voltado para experimentos em ambientes distribuídos. O arma-

zenamento e a consulta à proveniência empregam uma solução de forma centralizada a partir de uma base de dados em Prolog, entretanto, a manipulação de um grande volume de informação em uma base Prolog pode impactar no desempenho das consultas.

O diferencial da proposta do E-SECO ProVersion, objeto deste trabalho, em relação aos mencionados anteriormente, está relacionado ao acesso do pesquisador a repositórios de *workflows*, onde são armazenados os dados consumidos e gerados pelo *workflow*, através do uso de funcionalidades oferecidas pelo E-SECO ProVersion, auxiliando no processo de análise dos dados do experimento, bem como o *workflow* foi mantido durante todo o processo de experimentação. Todas as informações são capturadas por meio de um serviço web e armazenadas no banco de dados modelado com o PROV, para consulta a proveniência.

Considerando os trabalhos em manutenção e evolução de software, que podem contribuir com este trabalho, pode-se citar alguns trabalhos de relevância. O ProM Framework (GÜNTHER; AALST, 2006) atua na extração de dados do processo de desenvolvimento de software, tanto nos repositórios de código, como na IDE (Integrated Development Environment) de desenvolvimento. O FRASR (Framework for Analyzing Software Repositories) (PONCIN; SEREBRENIK; BRAND, 2011) também é um framework para análise do processo de desenvolvimento em repositórios de dados e extração de informações em sistemas de controle de versão.

O GiveMe Views (TAVARES *et al.*, 2015) é um framework de extração de dados históricos, em três diferentes tipos de repositórios sobre o ciclo de vida do software e em ferramentas de desenvolvimento. Através dos dados coletados e analisados, auxilia os desenvolvedores nos processos futuros de manutenção e evolução de software. A TimeLine Matrix (NOVAIS; JÚNIOR; MENDONÇA, 2012) é uma ferramenta que permite a visualização da evolução de componentes de software. Através de uma matriz 3x3 apresenta até 9 versões de um mesmo elemento a ser analisado. Além disso, permite ao usuário comparar até 3 componentes diferentes de software, com o objetivo de analisar a tendência entre os componentes com base em um conjunto variado de métricas.

O SEAgle (CHAIKALIS *et al.*, 2014) é um portal para análise do esforço de evolução de software por meio da mineração de repositórios. Apresenta um serviço web para coleta e análise de evolução de software em repositórios Git (sistema de controle de versão

distribuído e de gerenciamento de código fonte).

O conhecimento existente e as técnicas para gerenciamento da manutenção e evolução de software podem contribuir para o gerenciamento de *workflows* científicos adaptando a metodologia do versionamento, análise de repositórios e dos dados históricos. O E-SECO ProVersion aborda os conceitos existentes em manutenção e evolução de software no tratamento de *workflows* científicos. Através de sua arquitetura, os dados de execução de um *workflow* são capturados e armazenados em um repositório de proveniências modelado com o PROV e, com isso, é possível analisar o versionamento, agrupamento, derivação e proveniência retrospectiva dos *workflows*, utilizando ontologias para inferência de conhecimento implícito.

Uma análise comparativa entre os trabalhos relacionados pode ser vista na Tabela 3.8, onde são destacados entre os requisitos de comparação, quais atendem por completo, parcialmente ou não atendem ao requisito. Foram analisadas quatro características entre os trabalhos relacionados: versionamento, agrupamento, repositório e proveniência. A utilização dessas características são baseadas nos trabalhos de (GOBLE *et al.*, 2010) (JUNG; BAE, 2006) (OGASAWARA *et al.*, 2008) (SILVA *et al.*, 2010) . Como apresenta a Tabela 3.8, nenhuma abordagem trata de versionamento, agrupamento, repositório de *workflows*, e da proveniência dos dados do experimento por completo.

3.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou uma revisão *quasi* sistemática da literatura com o objetivo de caracterizar a área de estudo. Foram exibidos os critérios de busca e os resultados obtidos, também foram listados os principais pesquisadores da área, as principais conferências e os temas que estão sendo pesquisados com maior frequência. Ao final foram listados os trabalhos relacionados, envolvendo trabalho que tratam das temáticas *workflow*, proveniência, manutenção, evolução e gerência de configuração.

No capítulo seguinte será apresentado a proposta deste trabalho, a arquitetura E-SECO ProVersion.

Tabela 3.8: Comparativos dos trabalhos relacionados

Trabalho	Versionamento	Agrupamento	Repositório	Proveniência
Grafos de *Workflow*	SIM	NÃO	NÃO	NÃO
Clusters de *Workflows*	SIM	SIM	SIM	NÃO
SiDiff	SIM	NÃO	NÃO	NÃO
myExperiment	PARCIALMENTE	SIM	SIM	NÃO
CrowLabs	SIM	SIM	SIM	NÃO
Camera II	NÃO	NÃO	SIM	SIM
P-GRADE	NÃO	NÃO	SIM	NÃO
SimiFlow	SIM	SIM	NÃO	NÃO
CollabCumulus	NÃO	SIM	SIM	SIM
PBASE	NÃO	NÃO	SIM	SIM
Karma	NÃO	NÃO	NÃO	SIM
PASOA	NÃO	NÃO	NÃO	SIM
myGrid	NÃO	NÃO	SIM	SIM
D-OPM	NÃO	NÃO	NÃO	SIM
D-PROV	NÃO	NÃO	NÃO	SIM
SciProvMiner	NÃO	NÃO	NÃO	SIM
Redux	NÃO	NÃO	NÃO	SIM
ProvManager	NÃO	NÃO	SIM	SIM
ProM Framework	NÃO	NÃO	SIM	SIM
FRASR	SIM	NÃO	SIM	SIM
GiveMe Views	SIM	NÃO	SIM	SIM
TimeLine Matrix	SIM	NÃO	SIM	NÃO
SEAgle	SIM	NÃO	SIM	NÃO
E-SECO ProVersion	SIM	SIM	SIM	SIM

4 A ARQUITETURA E-SECO PROVERSION

Este capítulo tem como objetivo apresentar a arquitetura E-SECO ProVersion, no contexto da plataforma de ecossistema de software científico E-SECO (FREITAS *et al.*, 2015) . A E-SECO ProVersion pode ser vista como uma aplicação relacionada à plataforma de ecossistema E-SECO, adicionando funcionalidades específicas para a manutenção e evolução do experimento no contexto do ecossistema. Para um melhor entendimento da E-SECO ProVersion, detalhamos a plataforma do E-SECO para, a seguir, apresentar os requisitos e arquitetura específicos da E-SECO ProVersion.

4.1 E-SECO

A concepção de *workflows* científicos é uma abordagem bastante utilizada no contexto de e-Science. Existem pesquisas voltadas para o gerenciamento e execução de experimentos baseados em *workflows* de forma isolada. No entanto, experimentos complexos envolvem interações entre pesquisadores geograficamente distribuídos, podendo caracterizar-se como laboratórios colaborativos (OLSON, 2009) e que demandam a utilização de grandes volumes de dados, serviços e recursos computacionais distribuídos. Este cenário categoriza um ecossistema de experimentação científica (FREITAS *et al.*, 2015) .

Para conduzir experimentos neste contexto, pesquisadores precisam de uma plataforma flexível, extensível e escalável. Durante o processo de experimentação, informações valiosas podem ser perdidas e oportunidades de reutilização de recursos e serviços desperdiçadas, caso a plataforma de ecossistema para e-Science não considere estes aspectos. Neste contexto, foi proposta uma plataforma baseada em ecossistema de software, denominada E-SECO (E-Science Software ECOsystem) (FREITAS *et al.*, 2015) .

A E-SECO pode ser definida pelas suas relações com fornecedores de software científico, institutos de pesquisa, pesquisadores, órgãos de fomento, instituições financiadoras, e as partes interessadas nos resultados de pesquisa. Portanto, a plataforma (Figura 4.1) deve ser flexível, uma vez que ela pode integrar com plataformas científicas externas, que evoluem de maneira independente, e estão em constante evolução.

Estes relacionamentos ocorrem para gerar maior valor para o ecossistema, os quais requerem a abertura de suas fronteiras, onde aplicações terceiras passam a se conectarem e se beneficiarem de seus serviços, gerando valor para as partes envolvidas. Portanto, a plataforma do E-SECO precisa ser extensível. A E-SECO, além de ser provedora de serviços, é também uma consumidora de serviços de software científico, sendo necessário que a plataforma esteja apta a realizar novas integrações sem que haja modificações substanciais na solução. Finalmente, a plataforma precisa ser escalável, uma vez que ela é extensível, podendo ocasionar em um crescimento repentino e inesperado de requisições pelos serviços (FREITAS *et al.*, 2015) , principalmente com a utilização de laboratórios colaborativos.

Considerando o ciclo de vida de um experimento, a E-SECO possui um ciclo de vida baseado no ciclo de vida proposto por (BELLOUM *et al.*, 2011) , ilustrado na Figura 2.10, apresentando fases para a condução do experimento científico, tais como a revisão da literatura, por meio do Parsifal (FREITAS *et al.*, 2015) e Mendeley (HENNING; REICHELT, 2008) , ambas ferramentas externas de revisão da literatura integradas ao ecossistema. Na fase de prototipação do experimento é possível o acesso a diversos repositórios de serviços e *workflows*, como por exemplo o myExperiment (GOBLE *et al.*, 2010) e o BioCatalogue (BHAGAT *et al.*, 2010) . Já na fase de execução do experimento permite a transformação do *workflow*, a fim de viabilizar a execução em vários gerenciadores de *workflow* científico, como o Taverna (OINN *et al.*, 2007) , Kepler (ALTINTAS *et al.*, 2004) e Vistrails (FREIRE *et al.*, 2006) por meio de uma linguagem de interoperabilidade entre os SGWfC (BASTOS; BRAGA; GOMES, 2015) e a fase de análise dos resultados produzidos na execução do experimento.

No entanto, apesar da plataforma E-SECO representar um grande avanço em relação ao suporte necessário a um experimento científico em suas diversas fases, ela não apresentava os recursos necessários para o suporte a manutenção e evolução do experimento, controle da gerência de configuração e do controle dos dados de proveniência gerados em cada ciclo de experimentação.

Uma abordagem que considere tais requisitos do ciclo de vida de um experimento científico pode contribuir para o reuso dos *workflows*, serviços, configurações e resultados do experimento, provendo maiores informações sobre os dados de entrada e saída, os

recursos utilizados por cada *workflow* relacionado e o histórico do ciclo de vida. Assim, o objetivo da E-SECO ProVersion é adicionar recursos a plataforma E-SECO, de forma a gerenciar o ciclo de vida do experimento por meio das informações de proveniência capturadas, objetivando que informações valiosas e oportunidades de reutilização dos recursos e serviços não sejam perdidas por falta de controle na gestão do experimento.

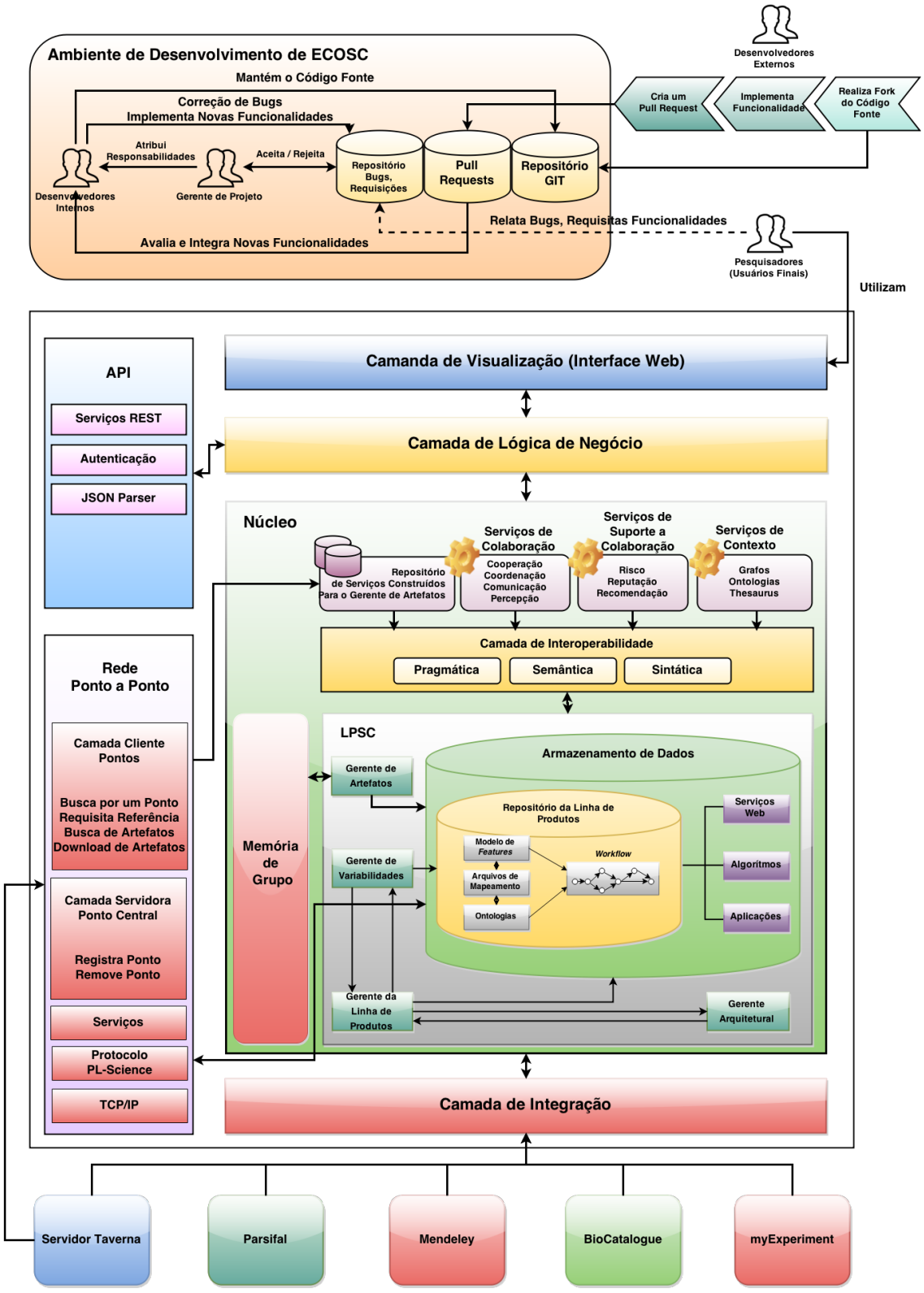


Figura 4.1: Plataforma E-SECO (FREITAS *et al.*, 2015) .

4.2 E-SECO PROVERSION

O propósito deste trabalho é tratar da gerência de configuração de um experimento e como este é manutenido e evoluído ao longo do seu ciclo do experimento, com base em seus dados históricos. Esse conhecimento, em conjunto com os dados utilizados, tais como configuração inicial, versão inicial do *workflow* e os resultados obtidos pelo pesquisador, são importantes para o reuso e replicação do experimento, uma vez que valida o mesmo.

Considerando este contexto, cada fase do ciclo do experimento científico apresenta tarefas específicas, e cada modificação da forma de execução dessa tarefa gera novas configurações para o experimento e possivelmente novas versões dos *workflows* que estão em uso no mesmo. Isso aplicado ao uso de laboratórios colaborativos deve ser muito bem controlado, para que os pesquisadores que estão interagindo sobre um mesmo experimento, em muitos casos com interação exclusivamente virtual, não se percam e tenham controle sobre os dados consumidos, processados e produzidos no experimento. Para isso, o experimento e consequentemente os *workflows* utilizados devem ser caracterizados como: em planejamento, em desenvolvimento, em revisão, em teste, em execução e em desuso.

Assim, considerando a necessidade da gerência da configuração do experimento e dos *workflows* por ele em uso, o ciclo de experimentação da E-SECO foi expandido para englobar a arquitetura E-SECO ProVersion (Figura 4.2). Na Figura 4.2, duas novas etapas foram adicionadas, a saber, a Gerência de Configuração e a Gerência de Proveniência. O controle da captura dos dados é realizado pela Gerência de Proveniência e o controle do experimento e dos *workflows*, pela Gerência de Configuração, sendo que estas etapas acompanham todo o processo. O detalhamento destas etapas será apresentado ao longo deste capítulo.

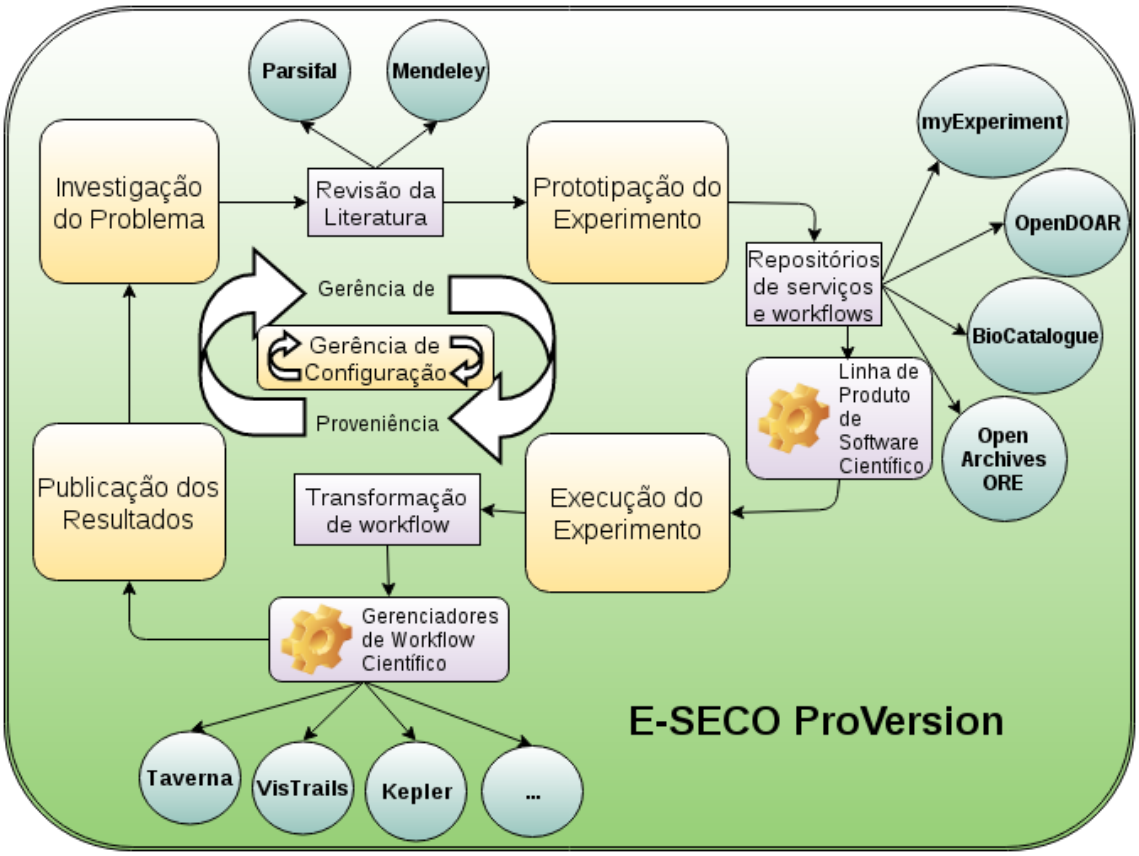


Figura 4.2: Ciclo da E-SECO com a expansão E-SECO ProVersion.

4.3 GERÊNCIA DE CONFIGURAÇÃO

A gerência de configuração deve ser entendida como uma etapa importante para o ciclo de vida de um experimento e dos *workflows* vinculados, pois, através dela, pode-se determinar o estado do experimento em um determinado momento, o que, considerando o ciclo de experimentação científica no contexto de laboratórios colaborativos, é de suma importância para o cientista, pois resultados parciais, erros encontrados ou gerados, entre outros, durante uma execução do experimento, podem derivar informações importantes para acertos futuros. Além disso, permite-se prever como o experimento tende a se comportar no futuro e no caso dos *workflows*, como os mesmos deve ser mantidos e evoluídos.

Conforme já dito, um *workflow* é um modelo computacional de um processo do mundo real e descreve todos os passos para realizar uma determinada etapa de um experimento. Contudo, modificações ocorrem a todo momento no mundo real e são refletidas nestes *workflows*, seja por mudanças em tarefas, serviços, ambiente ou mesmo no experimento em que ele se aplica, tornando a atividade de manutenção e evolução algo constante no

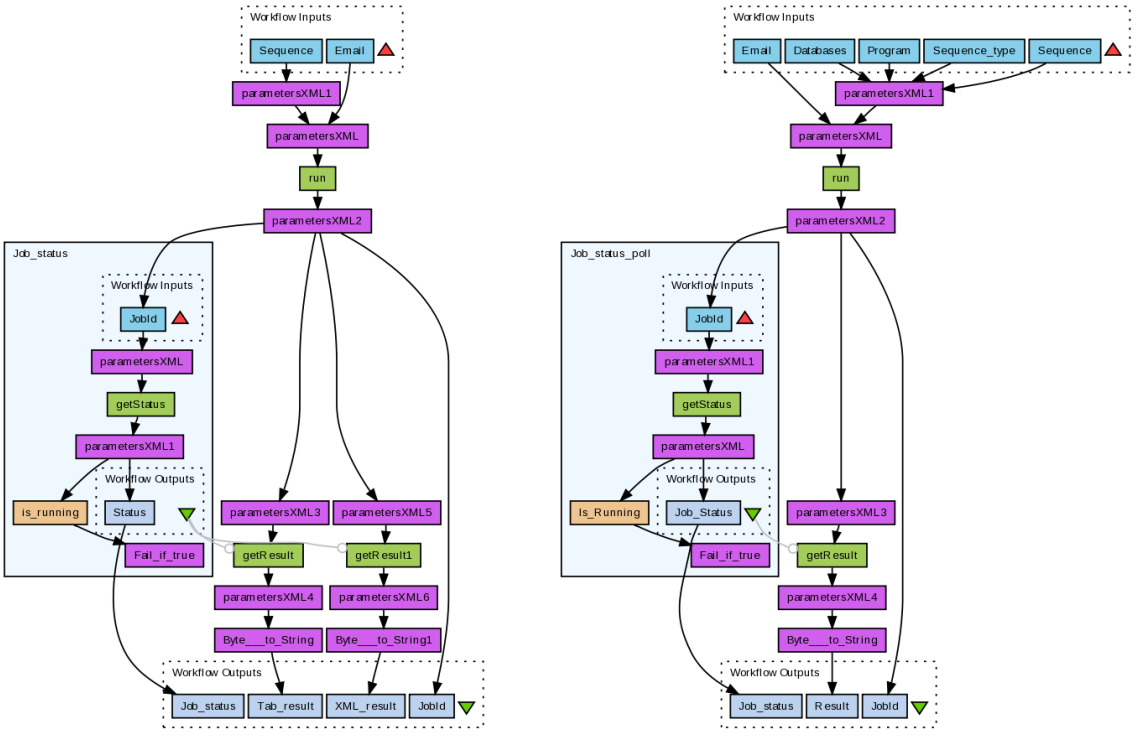
ciclo de vida do experimento e devendo estas modificações serem registradas e controladas pela gerência de configuração.

Conforme detalhado em (OGASAWARA *et al.*, 2008) , o apoio a reutilização e a gerência de configuração deve ser tratado como um item importante a ser explorado, de forma a diminuir o retrabalho e apoiar o aumento de produtividade e de qualidade dos experimentos dos pesquisadores. Neste contexto, soluções para a reutilização de *workflows* não podem depender apenas de um repositório, cujo principal suporte seja o download e upload de *workflows*, como é o caso do repositório myExperiment (GOBLE; ROURE, 2007) . São necessárias soluções mais abrangentes, que permitam tratar as manutenções e evoluções destes *workflows* ao longo do seu ciclo de experimentação e da gerência de configuração para permitir a reutilização do próprio experimento, considerando a inserção do *workflow* no mesmo. Além disso, um experimento pode ser formado por vários outros *workflows* ou *sub-workflows* em sua composição. Um exemplo de *sub-workflow* pode ser visto na Figura 4.3 (a[1] e b[2]), onde ambos os *workflows* apresentam em sua composição um *sub-workflow* com o mesmo fluxo de tarefas, com isso, uma necessidade de alteração no serviço deste *sub-workflow*, implicaria na manutenção de dois *workflows*.

Um experimento cientifico pode utilizar-se de vários *workflows* em diferentes SGWfC durante o ciclo de experimentação, sejam em etapas distintas ou em paralelo, podendo ocorrer a comunicação entre estes *workflows*. Estes *workflows* e a comunicação entre eles devem ser acompanhadas e controladas pelos pesquisadores para que o experimento sejam bem sucedido.

Para esboçar o ciclo de experimentação em um cenário similar, a Figura 4.4 apresenta 3 pesquisadores dispersos geograficamente, e cada um executa *workflows* distintos em diferentes etapas do experimento. Ao final, todos os dados consumidos, processados e produzidos sobre o experimento estão disponíveis na E-SECO ProVersion, o que permite a gerência dos dados, verificações necessárias, erros ocorridos, entre outros.

[1]http://www.myexperiment.org/workflows/1697.html
[2]http://www.myexperiment.org/workflows/1689.html



(a) (b)

Figura 4.3: Exemplo de *sub-workflow*.

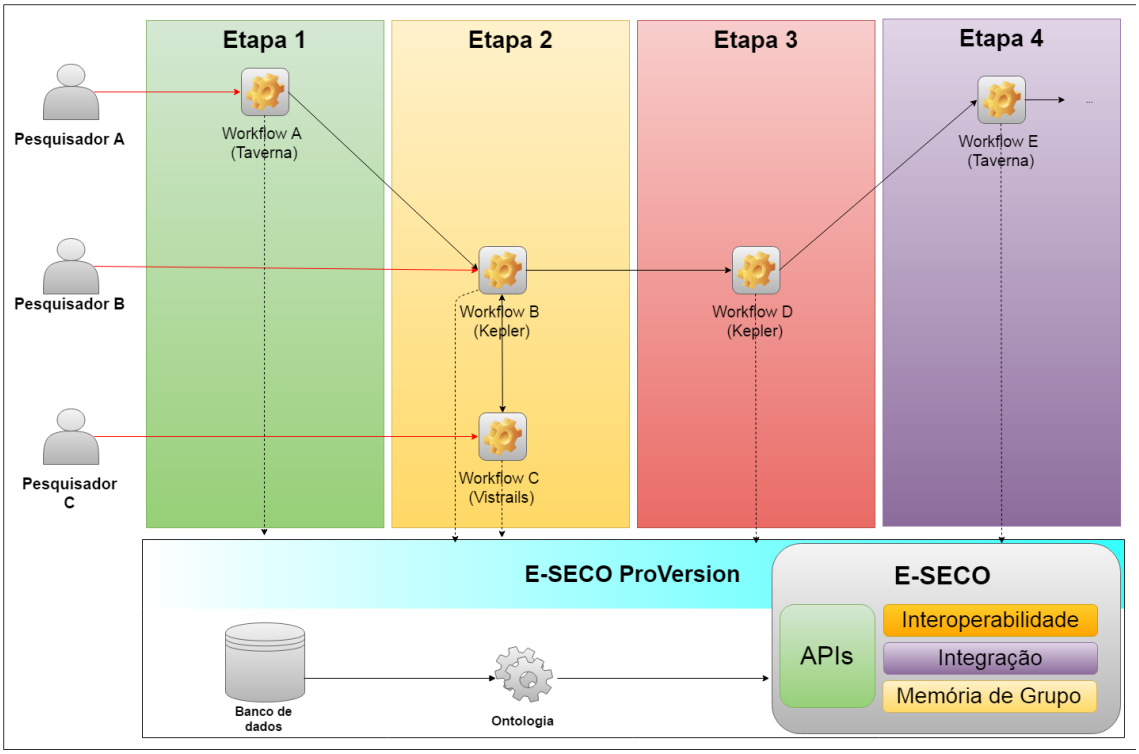


Figura 4.4: Acompanhamento do ciclo de experimentação.

Assim, acreditamos que para o pleno conhecimento sobre o experimento, o acompanhamento do comportamento dos *workflows* ao longo do ciclo de experimentação contribui para que informações valiosas do experimento não sejam perdidas e para isso, a proposta E-SECO ProVersion utiliza-se dos conceitos de gerência de configuração para o devido controle dos *workflows* e do experimento.

Na proposta E-SECO ProVersion, a Gerência de Configuração está dividida em duas etapas, Gerência de Proveniência e Gerência da Manutenção e Evolução, que serão detalhadas a seguir.

4.3.1 GERÊNCIA DE PROVENIÊNCIA

Um dos problemas dos processos de experimentação que utilizam *workflows* é a perda de conhecimento do pesquisador sobre o experimento, por conta do processamento computacional (MARINHO; WERNER; MURTA, 2009) , o qual executa ações que não são documentadas adequadamente, e dos SGWfC, que, na maioria dos casos, limitam-se a gerenciar as execuções dos *workflows* científicos de forma isolada do experimento ao qual faz parte. Geralmente, os SGWfC não registram as atividades de cada um dos *workflows* pertencentes ao experimento, suas ligações e nem a transformação dos dados que ocorrem entre cada uma dessas atividades durante a execução (DIAS, 2013) . Assim, para representar e apoiar o desenvolvimento do experimento científico de forma adequada, é necessário o registro das variações dos *workflows* associados a um experimento, visto que os mesmos são modificados no decorrer da pesquisa (MATTOSO *et al.*, 2009) . Este registro é também necessário para que estes *workflows* possam continuar a serem utilizados na pesquisa, pois a falta de informações adequadas pode causar um descontrole na condução do experimento, visto que o pesquisador desconhece a origem dos dados que foram gerados na execução, através de qual *workflow* foi gerado, e mais especificamente, de qual versão deste *workflow*, o que leva a um descontrole na gerência de configuração do experimento.

Geralmente, os pesquisadores que utilizam *workflows* científicos trabalham em um campo específico de investigação e não possuem um treinamento adequado. Com isso, muitas vezes começam uma aplicação copiando um *workflow* existente e, em seguida, ajustam o mesmo às novas necessidades de uso (COSTA *et al.*, 2015) , utilizando uma

abordagem descendente para a especificação dos novos *workflows*. Uma das formas de controlar o processo de experimentação é gerenciando a derivação dos *workflows* científicos durante a composição de um experimento, o que favorece o uso de uma abordagem descendente para a especificação de novos *workflows* para um novo experimento.

A abordagem descendente pode ser entendida como o histórico de evolução do *workflow*, através da qual o pesquisador pode compor novos *workflows* com base na reutilização de modelos anteriores. No entanto, a falta de informações sobre o *workflow* faz com que o pesquisador se perca no controle dos dados gerados, por desconhecer sua origem e o histórico de informações do mesmo. Como (HASAN; SION; WINSLETT, 2007) apresentam, o uso de ferramentas independentes do SGWfC para gerenciar o experimento e analisar os dados produzidos, faz-se necessária visto que parte dos SGWfC não provêm essa capacidade. Além disso, o controle da gerência de configuração dos experimentos e dos *workflows*, bem como o controle das manutenções e evolução que os *workflows* sofreram ao longo do experimento, tem que ser integrados ao ciclo de vida do experimento, conforme apresentado na Figura 4.2.

Assim, para tratar deste gerenciamento em um processo de experimentação, propõe-se o uso de um modelo de proveniência para coleta dos dados dos *workflows*, gerados durante sua modelagem e execução, visto que os mesmos possuem influência direta sobre a gerência de configuração do experimento. Como a E-SECO ProVersion está inserida no contexto de um ecossistema e a interoperabilidade dos dados é um fator importante, o uso de um modelo de proveniência padrão, amplamente aceito pela comunidade é importante. Neste sentido, o modelo PROV (MOREAU; MISSIER, 2013) foi o escolhido para ser utilizado na proposta desta dissertação. O modelo OPM (MOREAU *et al.*, 2011) também poderia ser utilizado neste contexto, no entanto, considera-se que o PROV seja uma evolução do modelo OPM, abrangendo muito mais relações causais e tendo um espectro de aplicação mais abrangente. O PROV auxilia na extração de conhecimento por meio de suas relações causais e, associado ao uso de ontologia, viabiliza a inferência de novo conhecimento.

Para exemplificar o funcionamento da gerência de proveniência, quando um determinado pesquisador, integrante de um laboratório colaborativo, executa um *workflow* e este está ligado a um experimento, todos os dados sobre o *workflow*, parâmetros de entrada, o fluxo entre as tarefas do *workflow* e os resultados produzidos, bem como as informações

sobre a qual experimento o mesmo faz parte, são coletados e enviados ao repositório da E-SECO ProVersion. Através da gerência de configuração, são extraídas as informações sobre o experimento e o *workflow*, e estas informações podem ser consultadas por todos os pesquisadores do laboratório, de forma a contribuir com a pesquisa sobre a ótica de controle dos dados do experimento e do *workflow*, permitindo o pesquisador tomar decisões estratégicas com relação aos próximos passos do estudo, como pode ser visto na Figura 4.5.

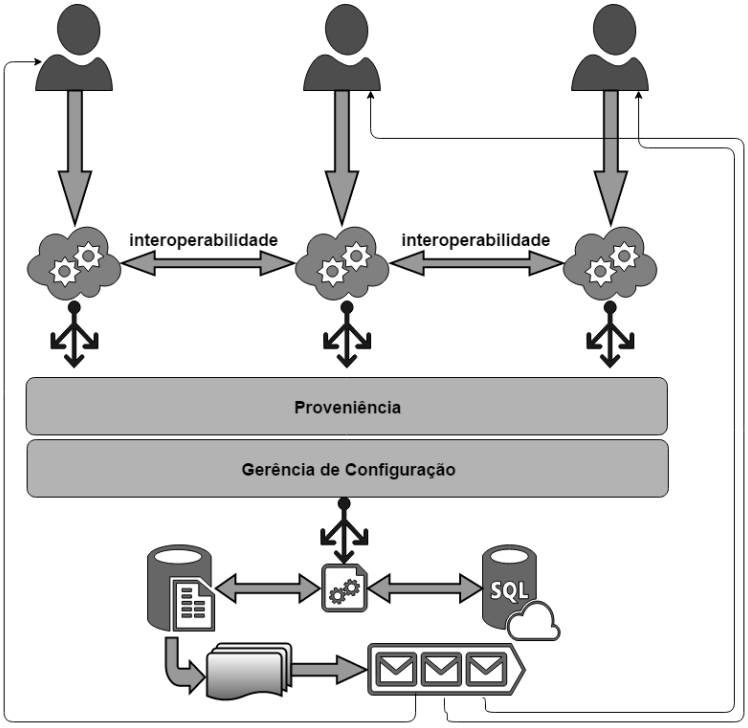


Figura 4.5: Captura dos dados e extração do conhecimento na E-SECO ProVersion.

4.3.2 GERÊNCIA DE MANUTENÇÃO E EVOLUÇÃO

A manutenção de software pode ser entendida como as modificações que ocorrem no produto após seu desenvolvimento ter sido concluído, seja para correção de erros, otimização ou adaptação a um novo ambiente. Já a evolução de software tem por objetivo entender como o mesmo é expandido e adaptado ao longo do seu ciclo de vida, onde é finalizado por seu decaimento, ou seja, não possui mais condições de operar.

Com isso, baseado nas ideias de evolução de software propostas por Lehman (Lehman (1980) e Lehman (1998)), nas abordagens de ciclos de vida dos *workflow* de (HOLLINGSWORTH *et al.*, 2004) , (DEELMAN; GIL, 2006) , (OINN *et al.*, 2007) ,

(HOLL *et al.*, 2014) e na classificação de manutenção de (SOMMERVILLE, 2011) , foi definida a proposta deste trabalho e a forma de gerenciamento do processo de evolução e manutenção de *workflows* e experimento.

A proposta baseia-se na construção de um modelo que armazene informações sobre mudanças do *workflow* e do experimento ao longo do ciclo de experimentação, capturando os dados através de um modelo de proveniência de dados. Dentre as manutenções aplicáveis destacam-se:

- Corretiva: as manutenções corretivas são para ajustes durante o processo de criação do experimento. Esse tipo de manutenção ocorre até que o processo de experimentação e os *workflows* estejam estáveis, permitindo sua utilização pelos pesquisadores de forma eficaz e eficiente. Essa manutenção tem como objetivo permitir que sua utilização continue ativa como, por exemplo, substituição de um serviço web por outro que apresente os mesmos recursos. Como Belhajjame *et al.* (2011) abordam em seu trabalho, as instituições que criam um *workflow* devem assegurar que os serviços constituintes sejam continuamente mantidos, o que torna-se um problema quando esse serviço é fornecido por terceiros. Nesse caso, uma falha em um *workflow* atinge diretamente o experimento, que também possivelmente acaba sendo modificado;

- Adaptativa: as manutenções adaptativas surgem da necessidade de executar um *workflow* ou experimento semelhante a um já existente, em outro SGWfC ou com novas características, assim utiliza-se como base um modelo disponível no repositório para a criação do novo *workflow* ou experimento;

- Evolutiva: as manutenções evolutivas ocorrem quando é necessário acrescentar algum recurso em um *workflow* ou experimento. Essa manutenção tem como objetivo expandi-lo, a fim de permitir que sua utilização continue ativa como, por exemplo, adição de outros serviços web, ou com novas informações baseadas nos resultados de um estudo anterior;

- Reengenharia: as manutenções de reengenharia podem ser classificadas como as de otimização dos processos do experimento ou do *workflow*. Esse tipo de manutenção trata de uma etapa constante no ciclo de vida do modelo proposto por Holl et al. (2012). Essa manutenção é importante, pois permite detectar a corretude e eficiência

na construção tanto do experimento como do *workflow*.

Assim, a cada nova versão, são gerados dados numa linha do tempo com o comportamento das evoluções a que o *workflow* foi submetido, em uma visão ao longo do tempo, conforme Figura 4.6. Essa visão da evolução dos *workflows* é importante pois impacta diretamente na gerência de configuração do experimento.

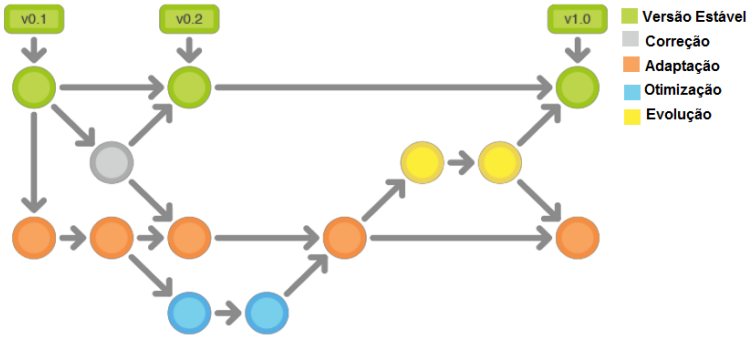


Figura 4.6: Versões do *workflow* ao longo do ciclo de experimentação.

Na Figura 4.6, os círculos em verde são as versões estáveis. Já os círculos em cinza podem ser caracterizados como as manutenções corretivas aplicadas ao experimento ou *workflow* durante o processo de experimentação. As manutenções adaptativas, representam a criação de um experimento ou *workflow* com base em um já existente, são representadas pelos círculos em laranja. Os círculos em azul referem-se a etapa de otimização do experimento ou *workflow*, conforme proposto por (HOLL *et al.*, 2014) . As manutenções evolutivas, dispostas pelos círculos em amarelo, podem abordar tanto o experimento ou *workflow* original como os derivados dele.

Considerando um cenário com *workflows* (Figura 4.7), um *workflow* desenvolvido por um Pesquisador A e disponibilizado em um repositório central, como por exemplo o myExperiment, pode ser usado por um Pesquisador B, o qual em seu experimento pode gerar n novas versões do *workflow* e aquelas consideradas estáveis no experimento, sejam compartilhadas novamente no repositório central, para que outros pesquisadores posam fazer uso das mesmas.

Para que este gerenciamento possa ser feito, é de suma importância conhecer a composição do *workflow*, quais tarefas e serviços utiliza, a ligação entre estes, o SGWfC em uso e com base nos dados de entrada, o resultado esperado de saída e a que tipo de experimento o mesmo se aplica, produzindo o conhecimento sobre as configurações do experimento, o

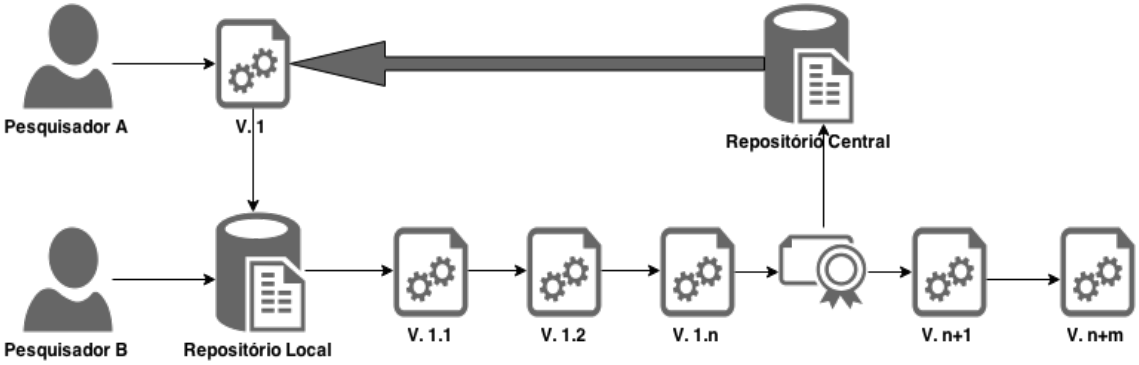


Figura 4.7: Troca de *workflows* entre repositórios e sua evolução.

qual esse trabalho se propõe.

4.4 ARQUITETURA

Considerando o novo ciclo de vida da E-SECO e as questões relacionadas à gerência de configuração, apresentadas na seção anterior, propõe-se a expansão da plataforma E-SECO (FREITAS *et al.*, 2015) , integrando uma nova arquitetura relacionada a gerência de configuração, conforme apresentado na Figura 4.8. Assim, para a gerência de configuração no contexto da E-SECO, foram propostos dois novos módulos, a saber, i) gerente de manutenção e evolução e ii) gerente de proveniência de dados.

O gerente de manutenção e evolução é um módulo do gerente de configuração no contexto da arquitetura E-SECO ProVersion, apresentando funções para o controle de dados de proveniência prospectiva e retrospectiva de um experimento, com o objetivo de controlar a evolução e as manutenções existentes no mesmo. O gerente de manutenção e evolução considera que o experimento pode ser composto de múltiplos *workflows* e para cada um são geradas diversas versões que devem ser controladas, além das informações do experimento a qual fazem parte.

O gerente de proveniência, também pertencente à gerência de configuração, é responsável pela coleta e armazenamento dos dados capturados durante a execução dos *workflows*, que são utilizados pelo módulo de manutenção e evolução. Estes dados contribuem para informar a necessidade de manutenção ou identificar uma evolução do mesmo. Erros de execução, problemas de desempenho, necessidade de substituição de recursos ou falhas em serviços, geram as informações de manutenção, sendo essas disponibilizadas para o pes-

quisador de forma clara e permitindo resolver tais problemas. O modo de funcionamento de cada um dos módulos é apresentado a seguir.

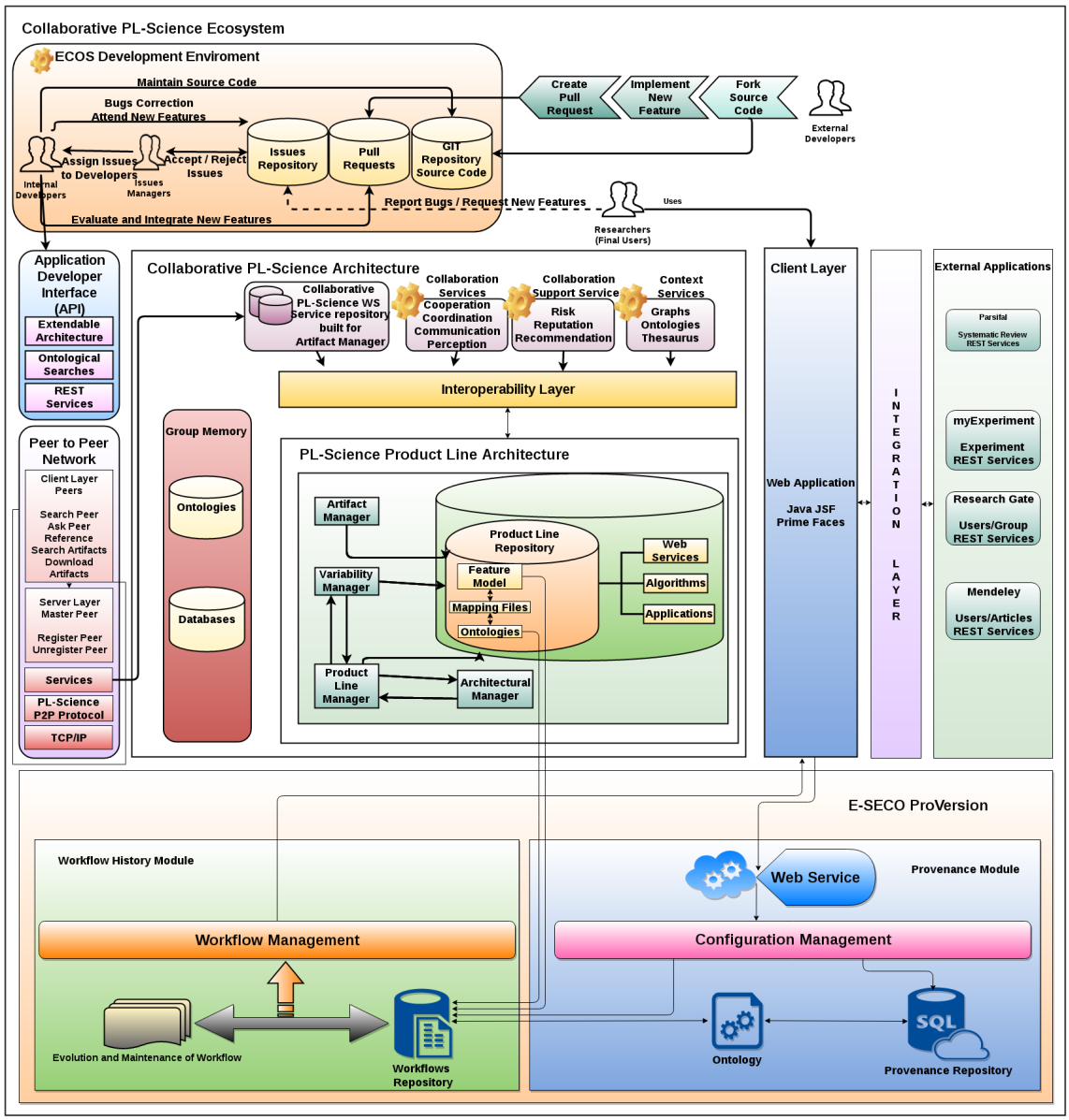


Figura 4.8: Arquitetura da E-SECO ProVersion (adaptada de (FREITAS *et al.*, 2015)).

4.4.1 MÓDULO DE PROVENIÊNCIA

Os dados gerados sobre o comportamento dos *workflows* ao longo de um experimento são coletados por um serviço web, que utiliza um banco de dados modelado de acordo com as regras de proveniência do modelo PROV. O uso de um modelo de proveniência auxilia na coleta, análise e distribuição dos dados, pois todas as informações como sua aplicação, parâmetros necessários para a execução, além do histórico de versões para cada *workflow*

ao longo do tempo, estão disponíveis no repositório junto as informações do experimento.

Além disso, a proveniência permite acompanhar todas as etapas do experimento, criando uma base de conhecimento sobre a pesquisa, fundamental em estudos onde existem laboratórios colaborativos (OLSON, 2009) . Essas informações de proveniência permitem que novos pesquisadores, ou mesmo pesquisadores que vinham utilizando um determinado *workflow*, possam conhecer as mudanças e, dependendo do contexto, migrar seu experimento para uma nova versão do *workflow* como, por exemplo, uma versão otimizada.

Além da coleta e o armazenamento no SGBD, uma ontologia integrada ao módulo é carregada com os dados do experimento para auxiliar na extração do conhecimento por meio de inferências, o que permite identificar *workflows* com tarefas similares, fluxos de trabalho próximos e características de manutenção e evolução no experimento e no *workflow*.

O modelo de dados utilizado pela E-SECO ProVersion, baseado no modelo PROV, bem como a ontologia especificada para derivação de novo conhecimento, são apresentados a seguir.

4.4.1.1 MODELO DE DADOS

O modelo de dados da E-SECO ProVersion estende o modelo de dados da E-SECO, agregando os recursos necessários para a gerência da evolução e manutenção do experimento e do *workflow*, seguindo o modelo de proveniência PROV. Conforme a Figura 4.9, os elementos destacados em vermelho representam os vértices principais e as relações causais existentes entre os vértices com base no PROV-DM (MOREAU; MISSIER, 2013) . Em verde estão os vértices criados especificamente para a E-SECO ProVersion seguindo as restrições do PROV-DM.

Todo o esquema do banco foi definido utilizando como base a língua inglesa seguindo as restrições do PROV-DM e preserva a estratégia de código aberto desenvolvido na E-SECO. Com isso não criam-se barreiras com desenvolvedores externos. A mesma estratégia foi adotada para o uso de um SGBD relacional, preservando a integridade e escalabilidade da solução. A descrição de uso de cada uma das tabelas do esquema apresentado na Figura 4.9, pode ser vista em detalhes no Apêndice A.1.

Os dados capturados a partir do serviço web acoplado à ferramenta de modelagem e execução do *workflow* permite a representação do conhecimento de proveniência do *workflow*, com a finalidade de utilização na identificação de evolução e de manutenções. Para exemplificar (Figura 4.3), ao instanciar um *workflow*, todos os dados por ele consumidos e gerados, além das tarefas e do fluxo entre elas, são enviados para a E-SECO ProVersion. Os dados armazenados no SGBD são carregados na ontologia que gera as informações a respeito das manutenções e evolução, seja do experimento ou dos *workflows*. Essas informações são posteriormente acessadas pelo modulo de manutenção e evolução e podem ser consultadas pelo pesquisador. Um exemplo dos dados inferidos na ontologia será apresentado na seção a seguir.

4.4.1.2 ONTOLOGIA

A adoção do modelo de proveniência, aliado ao uso da ontologia, o qual utiliza regras de inferência (Property Chains[3] em OWL 2.0) desenvolvidas para o propósito deste trabalho, facilita a descoberta de informações relativas a evolução e as manutenções de *workflows* ou experimento. Assim, com base nas informações obtidas utilizando as inferências, é possível identificar falhas ou modificações nas tarefas, permitindo visualizar os *workflows* que são afetados, bem como os experimentos a quais estão ligados, prevendo a necessidade de futuras manutenções.

A ontologia, denominada PROV-O (LEBO *et al.*, 2013) fornece um conjunto de classes, propriedades e restrições que podem ser usadas para representar e trocar informações de proveniência gerada em diferentes sistemas e contextos. Esta ontologia é púbica e disponibilizada pela W3C junto a especificação do PROV. No entanto, a PROV-O não expressa todo o conhecimento necessário para o suporte a evolução e manutenção de *workflows* e experimentos. O trabalho de expansão da ontologia PROV-O permite a descoberta de novas informações, tanto para a captura da proveniência prospectiva quanto retrospectiva.

[3]Property Chains surgiu no OWL 2 e funciona classificando objetos, onde permiti a transitividade entre múltiplas propriedades.

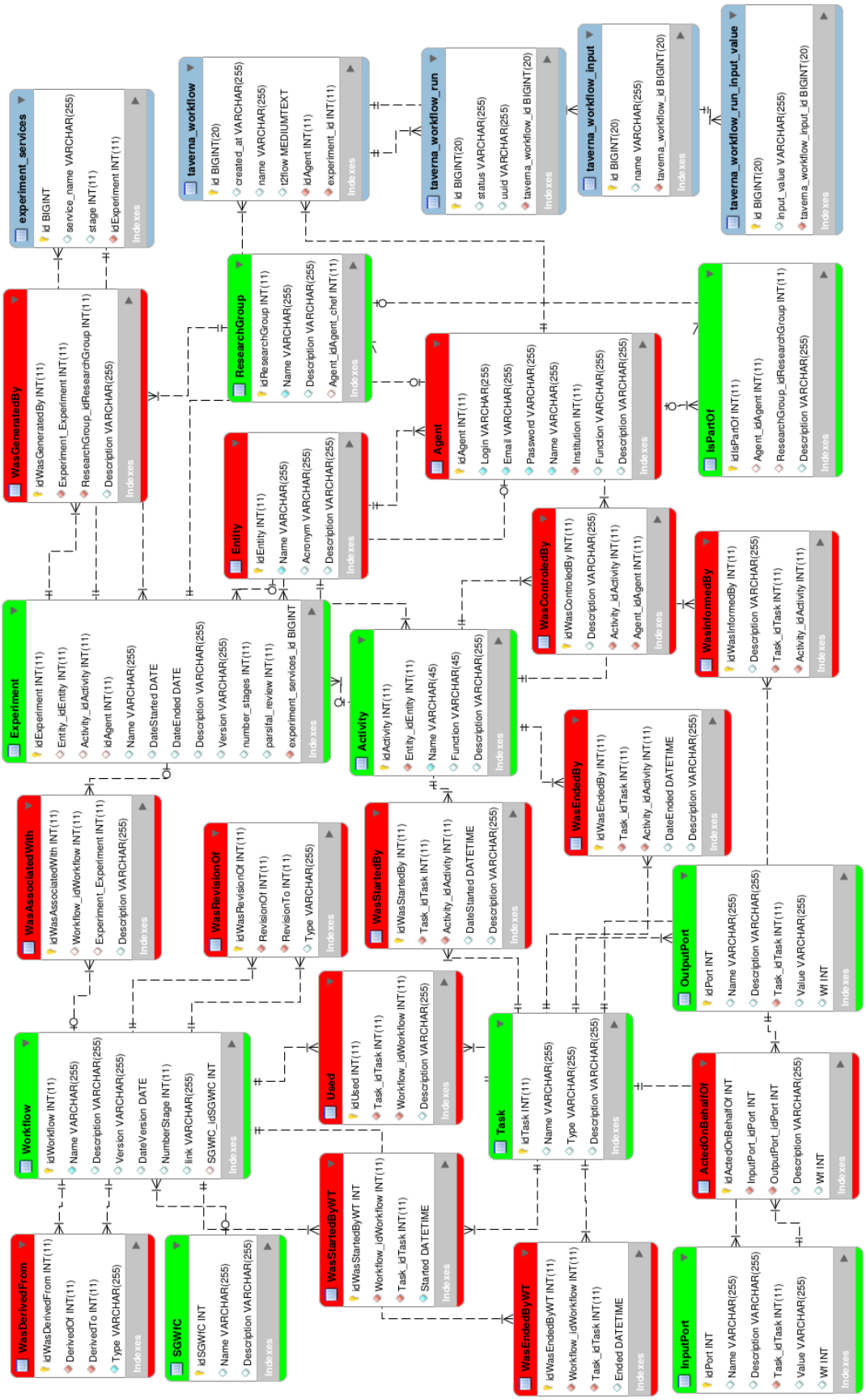


Figura 4.9: Esquema relacional do banco de dados.

A nova ontologia, denominada PROV-OEXT é alinhada ao modelo de dados proposto na seção 4.4.1.1, e permite a inferência e extração de novo conhecimento considerando as diferentes versões de *workflows* científicos. As relações causais da ontologia PRO-OEXT são exibidas na Figura 4.10.

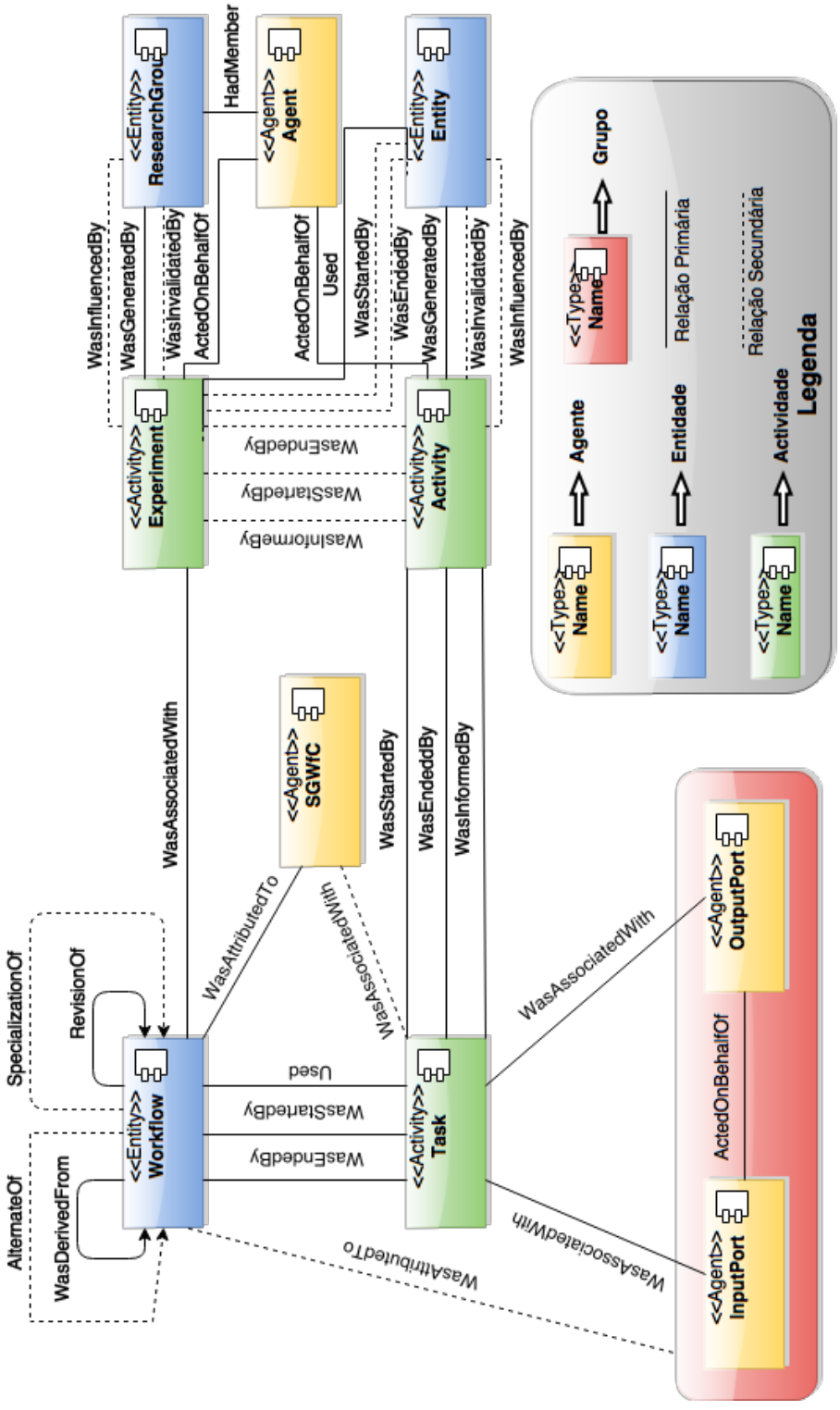


Figura 4.10: Relações causais da ontologia PROV-OEXT.

As principais mudanças realizadas na ontologia PROV-OEXT foram:

- Adição de novas classes:
  - 'Experiment' e 'Task', subclasses da classe 'Activity'. A primeira classe armazena informações dos experimentos registrado no E-SECO ProVersion e é utilizada na associação dos experimentos com os *workflows*. Já a segunda classe armazena as informações das tarefas disponíveis para compor *workflows* registrados no sistema.
  - 'InputPort' e 'OutputPort', subclasses da classe 'Task'. Ambas possuem como função registrar as portas de comunicação das tarefas utilizadas pelos *workflows*.
  - 'ResearchGroup', como subclasse de 'Person', sendo essa responsável por armazenar os grupos de pesquisas e posteriormente associa-los aos experimentos.
  - 'Workflow', subclasse de 'Entity', responsável por armazenar os dados dos *workflows* que são posteriormente interligados aos experimentos.
- Adição de novas propriedades (object properties):
  - A primeira representa a ligação entre 'Task' e 'SGWfC', conforme Figura 4.10, onde por meio da ligação com a classe 'Workflow' utilizando as object properties 'Used' e 'WasAttributedTo' inferiu-se a relação 'WasAssociatedWith'.
  - Outra inferência parecida ocorre entre as classes 'Workflow', 'Task', 'InputPort' e 'OutputPort', onde através das relações causais 'Used' e 'WasAssociatedWith' inferisse a relação 'WasAttributedTo'.
- Adição da property chain 'EvolutionOf', que classifica objetos entre múltiplas propriedades, sendo essas relações 'WasDerrivedFrom', 'SpecializationOf' e 'AlternateOf', e a relação inversa à está, denominada 'EvolutionTo'.

Além disso, a expansão da ontologia conta com as relações causais do PROV-O modificadas para permitir a inferência de conhecimento sobre evolução e manutenção de *workflows* e de configuração dos experimentos. A Figura 4.11 apresenta algumas informações inferidas pela ontologia, pois somente os dados do banco relacional não representam todo

o conhecimento disponível do *workflow*. Com essas informações, a E-SECO ProVersion pode sugerir mudanças estratégicas tanto na modelagem quanto na execução do experimento ou *workflows* associados, auxiliando o pesquisador ou o grupo responsável pelo experimento, na melhoria do mesmo e na identificação de pontos falhos ou que necessitam ser corrigidos.

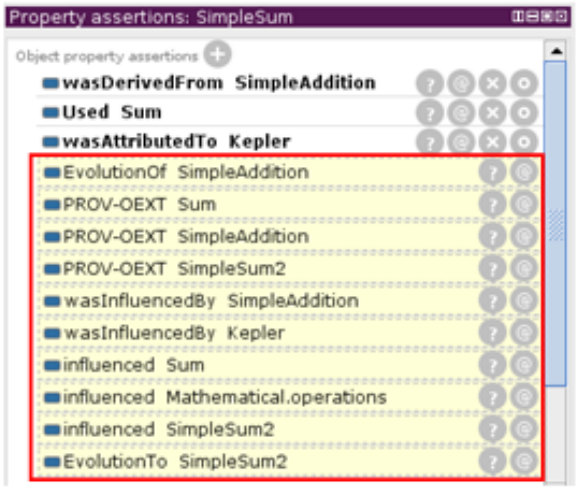


Figura 4.11: Inferências do PROV-OEXT.

A Figura 4.11 mostra que são inferidas informações sobre quais *workflows* influenciaram na criação do *workflow* analisado. No exemplo, que considera que o *workflow* foi executado no SGWfC Kepler, e através das Properties Chains 'EvolutionTo' e 'EvolutionOf', infere a linha evolutiva do *workflow* e quais outros *workflows* foram influenciados por ele. Essas informações são importantes para auxiliar na manutenção, evolução e reuso deste *workflow*.

4.4.2 MÓDULO DE MANUTENÇÃO E EVOLUÇÃO

O módulo de gerência de manutenção e evolução concentra as principais funcionalidades relacionadas a manutenção e evolução do experimento e dos *workflows*, permitindo ainda que os pesquisadores consultem os dados dos *workflows* utilizados em seus experimentos bem como seus parâmetros de entrada e saída, as tarefas, serviços que o compõem, informações sobre cada execução, as portas de comunicação utilizadas e os resultados produzidos. Este módulo trabalha integrado ao módulo de proveniência, de forma automatizada, associando *workflows* e experimentos.

Para apoiar a condução e agregar novo conhecimento ao experimento, a E-SECO ProVersion propõe alguns recursos, que englobam desde o conhecimento sobre o pesquisador até a proposta principal da gerência de configuração do experimento. Considerando a gerência das instituições, a E-SECO ProVersion utiliza-se dos cadastros das instituições, contendo nome, sigla e descrição, realizados de forma manual, para identificar as entidades que fazem uso do sistema e associar os pesquisadores dessas instituições, através de cadastro pelo próprio pesquisador, de forma que futuramente possam ser identificados grupos com interesses similares. Dados dos pesquisadores também são armazenados, incluindo nome, e-mail, instituição a qual está vinculado, função e uma descrição de seu perfil, conforme Figura 4.12. Essas informações são utilizadas para compor os grupos de pesquisa e vincular o experimento a um pesquisador responsável, permitindo listar os experimentos de cada pesquisador.

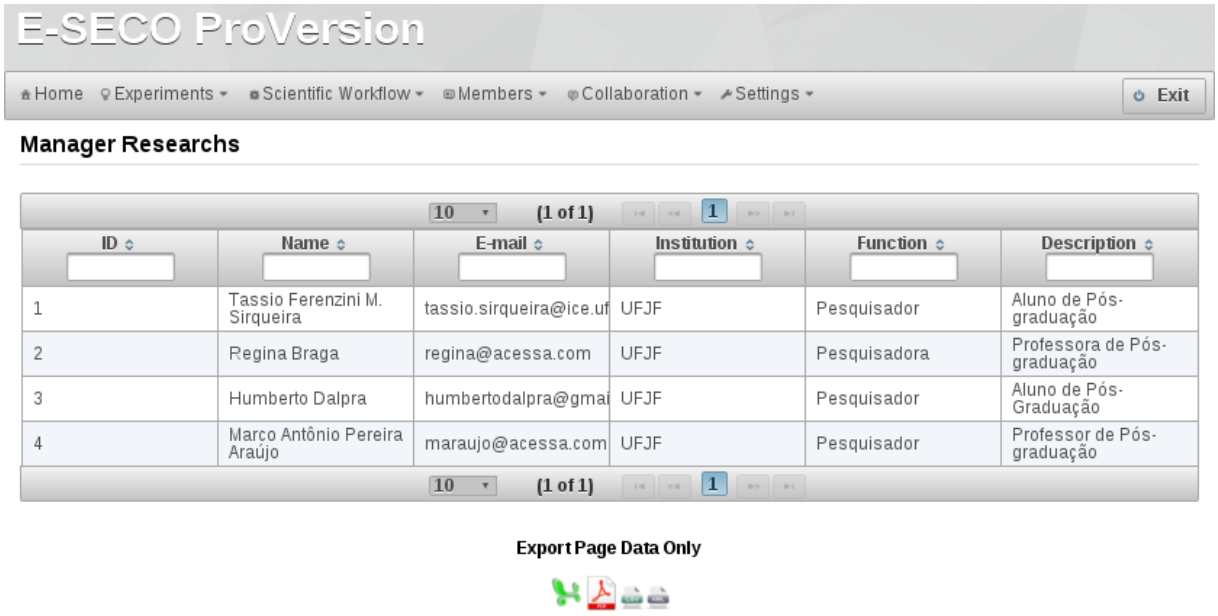


Figura 4.12: Gerência de pesquisadores.

Outro controle importante no contexto da gerência de configuração do experimento é a gerência dos grupos de pesquisas, que permite identificar os grupos de pesquisas existentes, e inclui informações como o nome do grupo, sua descrição de pesquisa e o pesquisador responsável por tal grupo. Assim, o pesquisador responsável pelo grupo pode gerenciar os pesquisadores que fazem parte deste grupo e estes podem administrar os experimentos criados pelo grupo, conforme Figura 4.13.

Figura 4.13: Pesquisadores por grupo de pesquisa.

No que diz respeito às informações do experimento, foram agregadas novas informações às já existentes na E-SECO. Informações como a data de início e término do experimento, entidade responsável, versão do experimento e as atividades pertencentes ao mesmo, no intuito de agregar conhecimento acerca do ciclo de experimentação. Outras informações vinculadas ao experimento são os *workflows* que foram utilizados no mesmo. Com isso, todos os *workflows* utilizados durante um processo de experimentação ficam registrados junto ao experimento, conforme Figura 4.14, evitando que esse conhecimento fique retido somente com o pesquisador que o instanciou ou seja perdido ao longo do experimento, além de facilitar a gerência de uma dada configuração do experimento, ou seja, o conjunto de *workflows* e dados utilizados em uma determinada execução do experimento.

Considerando os *workflows*, a arquitetura E-SECO ProVersion armazena todas as informações relevantes para seu posterior reuso, incluindo o SGWfC utilizado, bem como as tarefas disponíveis. As informações das tarefas permitem verificar os *workflows* que possuem compartilhamento das mesmas, o tipo de tarefa e a que ela se aplica, conforme Figura 4.15. Assim, para a composição de novos *workflows*, com base em uma linha evolutiva, os pesquisadores podem conhecer em detalhes o que será executado pelo *workflow*.

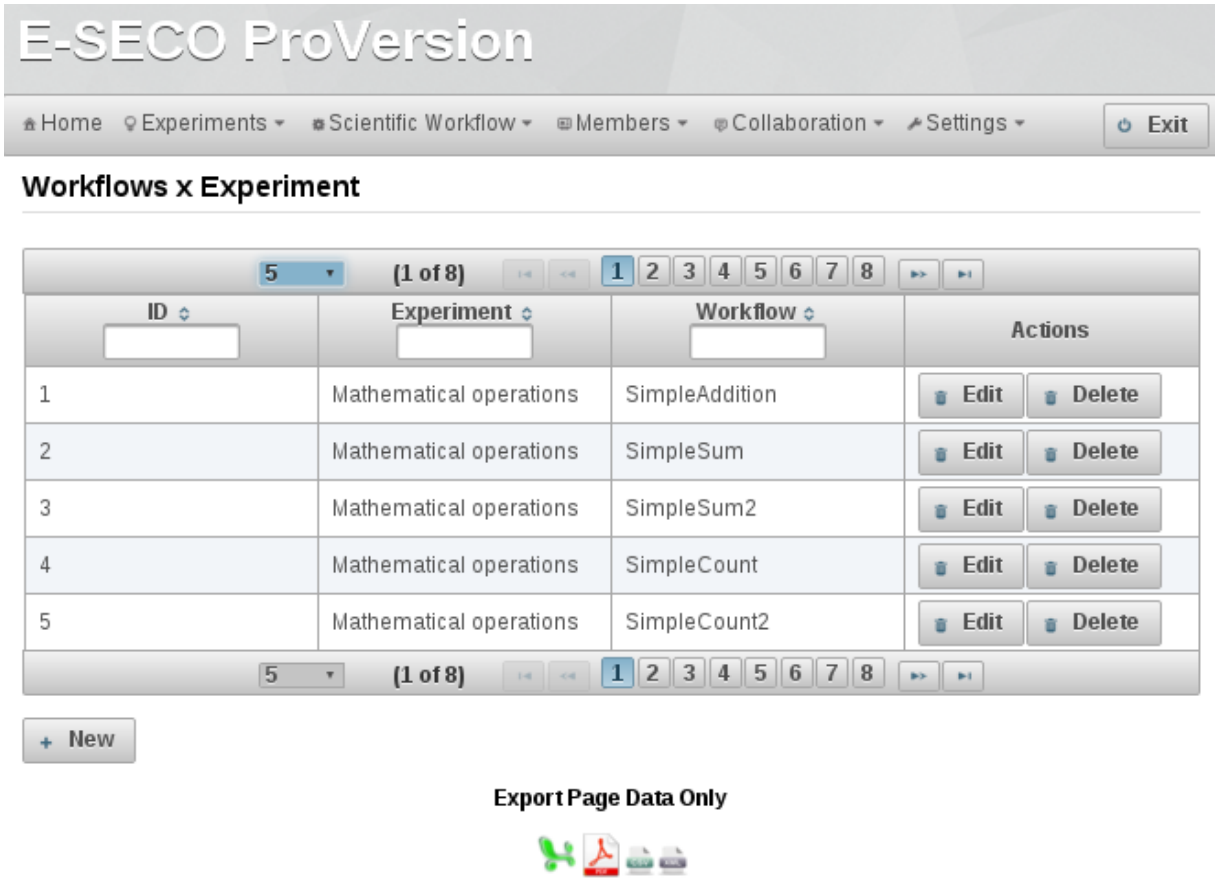


Figura 4.14: *Workflow*s associados aos experimentos.

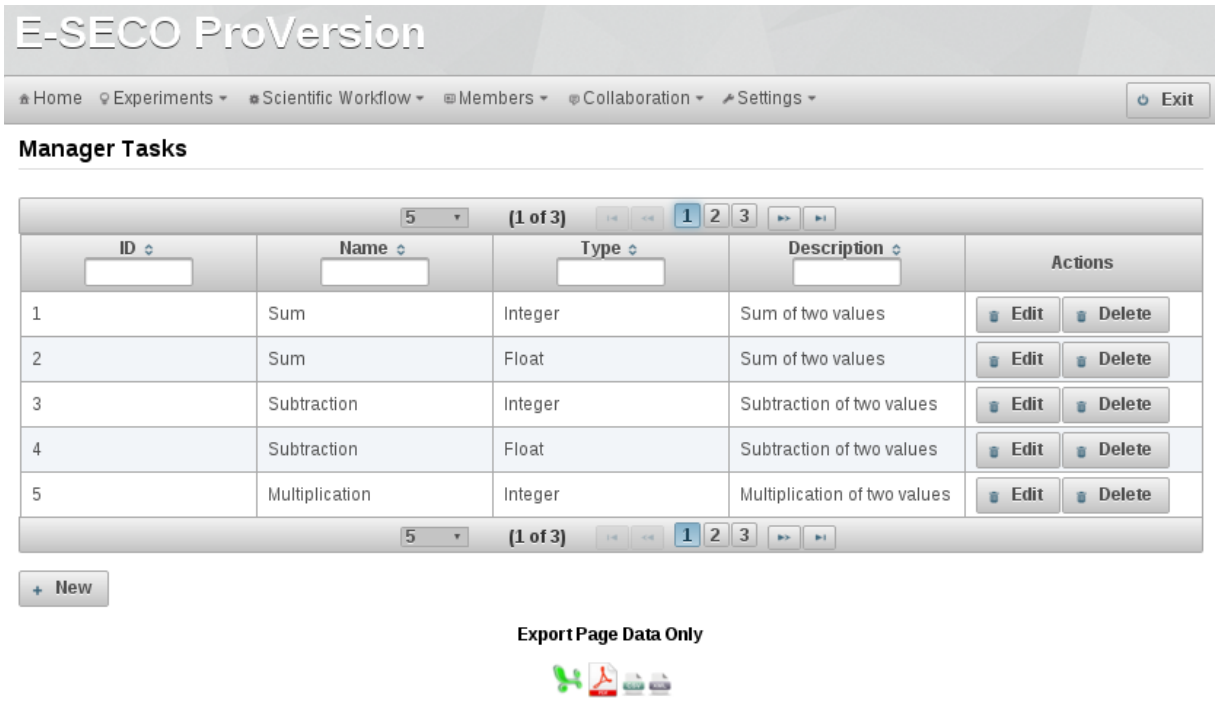


Figura 4.15: Informações das tarefas.

Tratar a manutenção e a evolução do experimento e dos *workflows* no contexto de um ciclo de experimentação é importante, uma vez que permite que estes experimentos e *workflows* relacionados continuem ativos e possam ser reavaliados, ou mesmo reutilizados, para compor outros estudos. Para que essa tarefa possa ocorrer de forma mais fácil e transparente para os pesquisadores, a E-SECO ProVersion fornece uma interface de gerenciamento dos dados do *workflow*, onde são registrados seu nome, descrição, versão, data da versão, número de estágios, link do repositório e o SGWfC do mesmo, conforme Figura 4.16. Esses dados compõem as informações básicas do *workflow*, pois além destes, a E-SECO ProVersion registra as tarefas que estão sendo utilizadas, com informações de descrição, registro das execuções com os dados de início e fim de cada uma, bem como as portas de entrada e saída utilizadas na comunicação, troca dos dados e o fluxo entre as tarefas. Estes dados podem ser considerados como dados de proveniência prospectiva, visto que apresentam os passos que serão executados pelo *workflow* ao ser instanciado, conforme Figura 4.17.

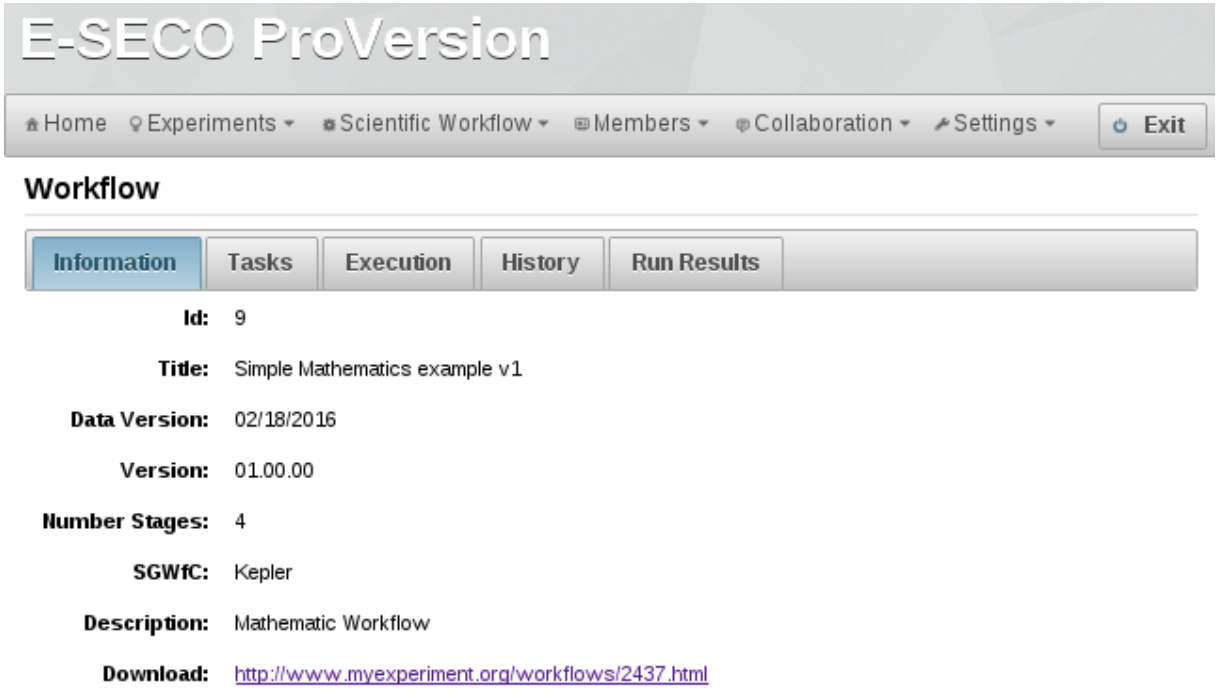


Figura 4.16: Informações do *workflow*.

Na E-SECO ProVersion, considerando o número de tarefas e a aplicação em um determinado *workflow*, é possível detectar quais *workflows* apresentam algum grau de similaridade, baseado nas tarefas em comum, conforme Figura 4.18. Também é possível obter informações sobre falhas em tarefas, através do histórico de execução, capturados

Figura 4.17: Informações das tarefas no *workflow*.

pelo módulo de proveniência e informações de mudanças entre as versões de um *workflow*, conforme Figura 4.19, sendo todos os dados capturados tratados pela ontologia no módulo de manutenção e evolução.

Considerando ainda estas funcionalidades, os dados coletados pelo modulo de proveniência durante a execução do *workflow*, são analisados com o uso da ontologia PROV-OEXT, permitindo extrair o histórico de evolução e manutenção. Um exemplo da interface contendo as informações inferidas da ontologia PROV-OEXT de um *workflow*, pode ser visto na Figura 4.20.

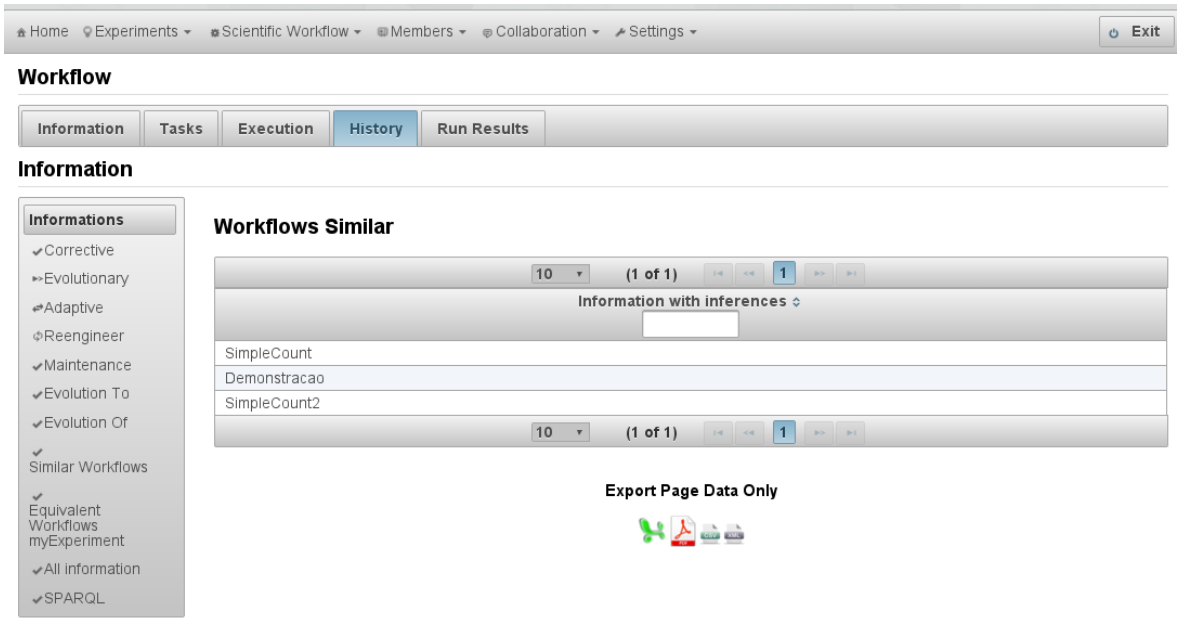


Figura 4.18: Similaridades entre *workflows* baseado em tarefas compartilhadas.

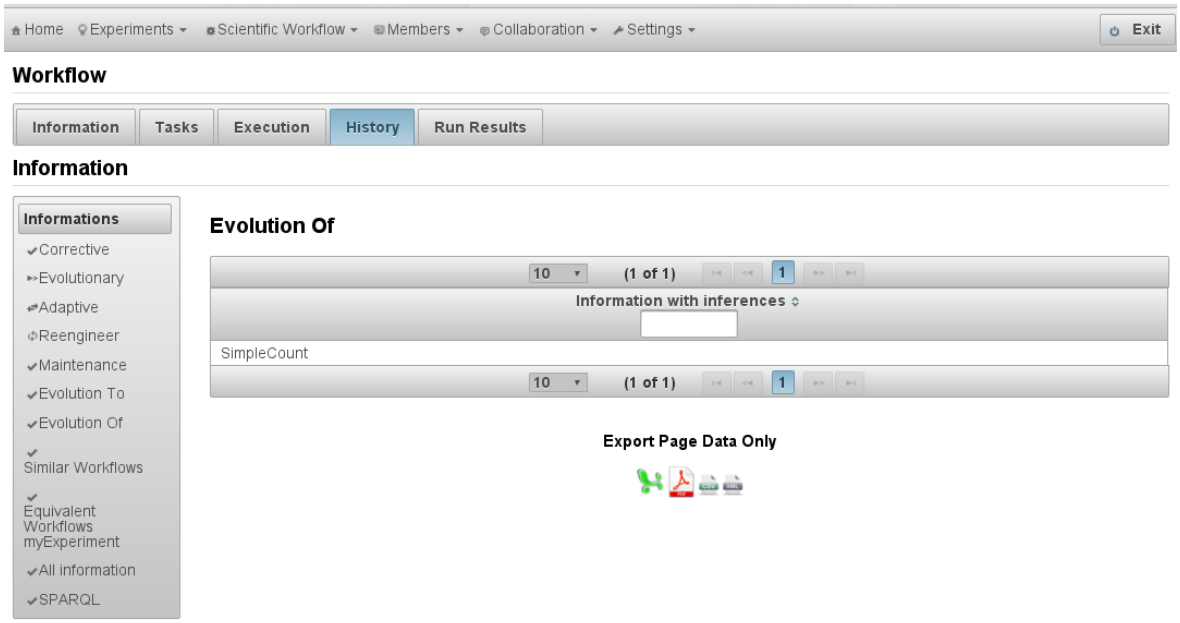


Figura 4.19: Informação de evolução do *workflows* extraída da ontologia.

Outra funcionalidade relacionada disponível na E-SECO ProVersion é a busca por *workflows* similares ao selecionado, no repositório myExperiment. Com isso é possível verificar no repositório quais outros *workflows* compartilham de tarefas similares, se possuem alguma característica de evolução, permitindo ao pesquisador optar por utilizar o *workflow* analisado ou qualquer outro disponível. Para a busca por *workflows* similares no repositório, a pesquisa é feita comparando-se *workflows* que possuem tanto o número de estágios quanto tarefas similares, conforme Figura 4.21.

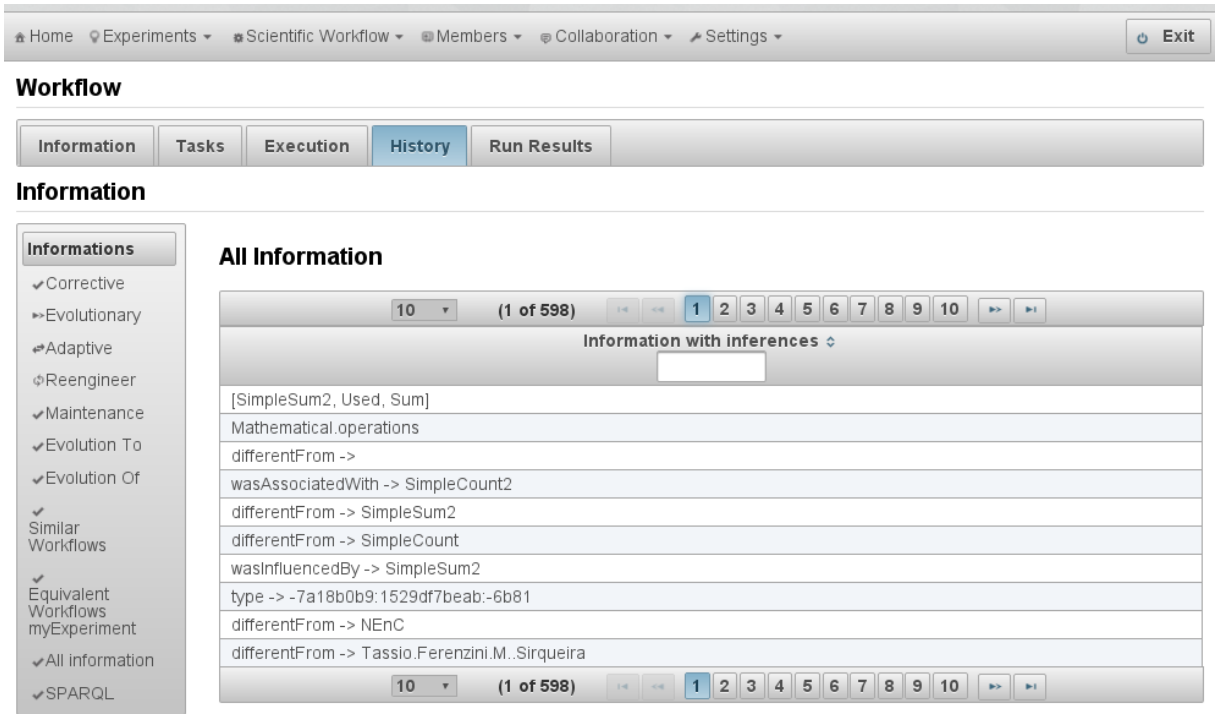


Figura 4.20: Interface com as inferências do PROV-OEXT.

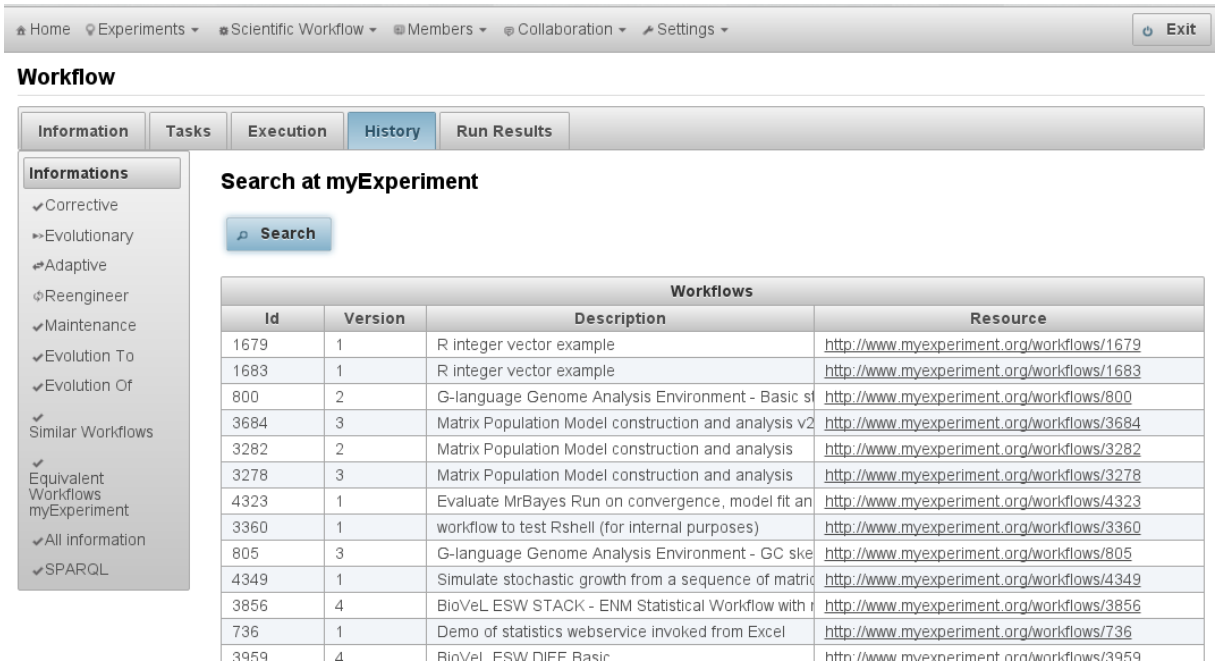
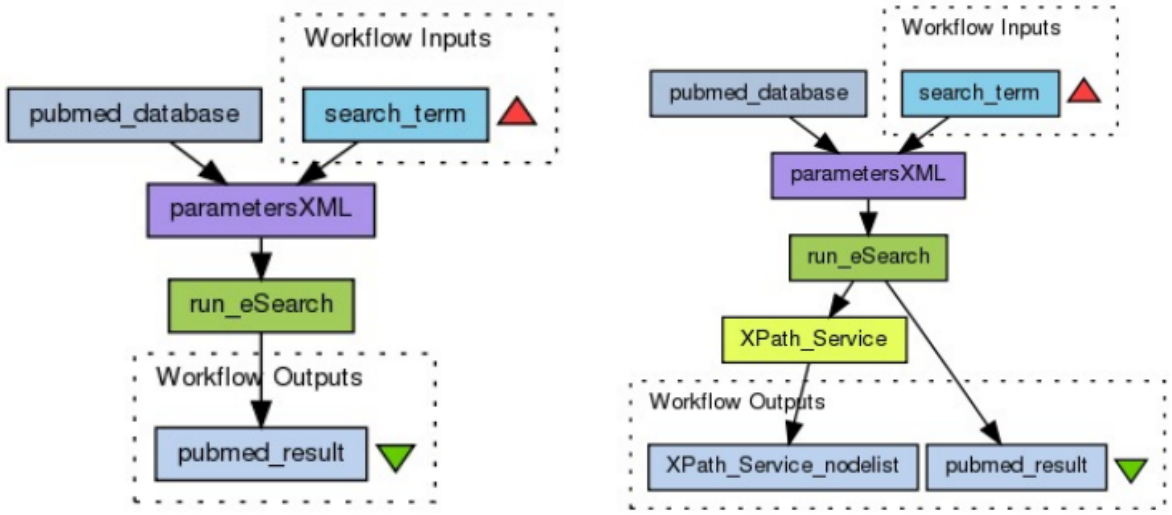


Figura 4.21: Interface da E-SECO ProVersion acessando o myExperiment.

Um exemplo de uso da E-SECO ProVersion é apresentado nas figuras 4.22 (A[4] e B[5]), com *workflows* criados pelo mesmo pesquisador (Paul Fisher) e que possuem como característica a busca de informações em uma base médica. Considera-se o *workflow* A

[4]http://www.myexperiment.org/workflows/1975.html
[5]http://www.myexperiment.org/workflows/1976.html

uma versão base para a criação do *workflow* B, e, com o uso da E-SECO ProVersion, outros pesquisadores que utilizam este *workflow* e que não possuem conhecimento sobre a existência de duas versões, podem ser informados sobre isso e podem utilizar-se destas, evitando o retrabalho, além de obter informações sobre seu sucesso ou falha, problemas reportados durante a execução, necessidades de modificação ou otimização entre outras possibilidades.



(a) (b)

Figura 4.22: *Workflow* B evoluído de A.

A Figura 4.22 esboça *workflows* do SGWfC Taverna, no qual as tarefas foram cadastradas junto a E-SECO ProVersion, gerado um identificador para cada tarefa. Considerando as tarefas que foram utilizadas, é possível detectar a quais *workflows* as mesmas estão vinculadas. O uso de uma mesma tarefa em mais de um *workflow* é uma das formas da E-SECO ProVersion verificar a similaridade entre os mesmos, podendo ser maior ou menor, dependendo do número de tarefas comuns a ambos. Apesar de existirem técnicas específicas para cálculo de similaridade como a apresentada por Yu e Huang (2016), neste trabalho foi aplicada a comparação entre nomes de tarefas, visto que a questão de similaridade envolve aspectos de interoperabilidade e que desvia do foco deste trabalho.

É possível notar que o *workflow* B é uma evolução do *workflow* A, ambos apresentados na Figura 4.22, haja vista que as entradas e, consequentemente, os parâmetros, são os mesmos. Também é utilizada a mesma tarefa 'run_eSearch' por ambos. Porém, o *workflow* B tem a adição de um novo recurso denominado 'XPath_Service', o que resulta em uma saída diferente do *workflow* A.

Estas informações ajudam a identificar a necessidade de uma manutenção corretiva, tal como a substituição do serviço antigo por outro similar, mantendo o *workflow* útil à pesquisa para a qual foi desenvolvido inicialmente ou para a utilização em outros experimentos. Um exemplo, disponível no repositório myExperiment, é apresentado na Figura 4.23[6], onde o mesmo possui serviços que não estão mais operantes, identificados durante a sua execução.

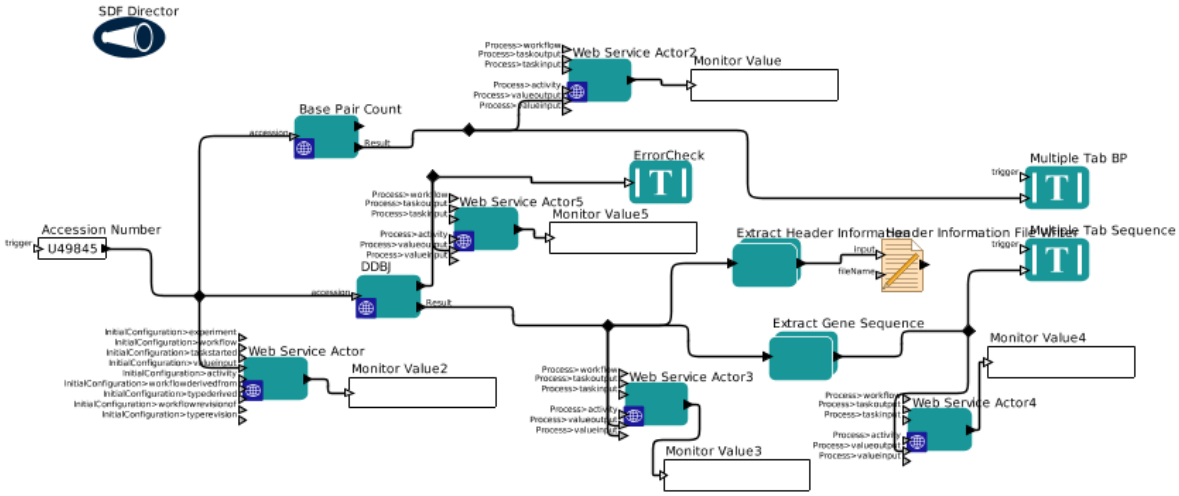


Figura 4.23: *Workflow* com serviço indisponível.

Todavia, considerando repositórios de *workflows* existentes, como o myExperiment, entre outros, é difícil encontrar informações sobre os *workflows* e principalmente sobre seu histórico de execução. A falta destas informações dificulta a análise dos resultados do experimento e a análise do ciclo de vida do *workflow*, o que pode impedir e/ou dificultar a reutilização deles em outros experimentos, visto que não se conhece claramente a origem do mesmo. Portanto, a criação de uma base de dados contendo o histórico do ciclo de vida do *workflow*, pode auxiliar no reuso e entendimento dos mesmos pelos pesquisadores. Desta forma, consideramos que a proposta de um meta-repositório com dados históricos de *workflows* pode trazer ganhos para o domínio de e-Science, melhorando a visibilidade e reutilização para *workflows* e experimentos.

[6]http://www.myexperiment.org/workflows/2459.html

4.5 ESTUDO PRELIMINAR DE REPOSITÓRIOS EXISTENTES

A E-SECO ProVersion possui como objetivo auxiliar os pesquisadores na proposição de melhorias e identificação de falhas em um experimento científico e seus *workflows* relacionados, através do uso de ontologias e inferências. No entanto, parte primordial para o aprimoramento dos resultados é a disponibilidade de dados históricos dos *workflows*. O uso de ontologias e mecanismos de inferência auxiliam na derivação de conhecimento implícito, para o aprimoramento da gerência de configuração do experimento e dos *workflows*, mas informações importantes também podem ser descobertas a partir dos dados históricos.

Considerando esse contexto, um dos problemas encontrados para análise histórica de *workflows* científicos e a proposição de melhorias com base nestes dados, é a falta de dados históricos que permitam realizar a análises relativas à evolução e manutenção, e com isso, propor melhorias com base nestes dados. Apesar de existirem repositórios com dados de *workflows* como o CrowLabs (MATES *et al.*, 2011) e o myExperiment (GOBLE *et al.*, 2010) , estes são incompletos, com informações inconsistentes e, na maioria das vezes, não muito organizados do ponto de vista de meta-dados. Além disso, para uma análise histórica mais abrangente, envolvendo os experimentos, até onde pesquisamos não se tem repositórios públicos com dados de pesquisa para avaliação. Assim, com o objetivo de criar uma base de conhecimento sobre o assunto, foi realizada uma pesquisa em repositórios de *workflows*, no intuito de levantar um conjunto de informações sobre o ciclo de vida dos mesmos. Os resultados são apresentados nas subseções subsequentes.

4.5.1 ANÁLISE DOS REPOSITÓRIOS DE *WORKFLOW*S

Com o objetivo de analisar a evolução e manutenção dos *workflows*, foi realizado um levantamento nos repositórios myExperiment (GOBLE *et al.*, 2010) e CrowLabs (MATES *et al.*, 2011) , sendo estes os únicos repositórios disponíveis no momento da pesquisa.

Na análise do repositório CrowLabs, identificou-se que o mesmo não possuía dados referentes a versões de um mesmo *workflow*, lista das tarefas utilizadas com suas respectivas descrições e nem os parâmetros de entradas para execução do mesmo, constando apenas os *workflows* para uso. Outro problema identificado neste repositório foi em relação à per-

missão de acesso aos *workflows* disponíveis. Não foi possível acessar qualquer *workflow* para reutilizá-lo, o que em um cenário colaborativo para reuso de *workflows* científicos é uma característica essencial.

Com relação ao myExperiment, apresenta informações sobre versionamento, com os detalhes dos *workflows* e permite que sejam acessados para seu reuso. Um fato a ser observado é com relação à quantidade e grau de informações sobre os *workflows* disponíveis, pois a grande maioria dos *workflows* possuem poucas informações documentadas, histórico de versão e de execuções disponíveis.

Considerando o cenário e a dificuldade de acesso aos repositórios, foi realizada uma análise somente do repositório myExperiment, por ser o único que possui dados minimamente suficientes para uma análise, no contexto de evolução e manutenção de *workflows*. Desta análise, esperamos que um conjunto de requisitos para a criação de um meta-repositório com dados históricos para evolução e manutenção de experimentos científicos possa ser criado.

4.5.1.1 ANÁLISE DO MYEXPERIMENT

Foi realizado um levantamento de dados no repositório myExperiment ao longo do mês de outubro de 2015 e constatou-se que entre os 3692 *workflows* disponíveis na base, 1571 utilizam os SGWfCs Taverna 2, Kepler ou Vistrails, sendo esses os principais SGWfC, representando mais de 40% do total, a outra parte estava distribuída em um número pequeno de *workflows*, sendo apenas o Taverna 1 que destaca-se com 566 *workflows*, mas que não é possível executar no SGWfC do Taverna 2. Destes, 1520 são *workflows* desenvolvidos no SGWfC Taverna, 47 no SGWfC Kepler e 4 no Vistrails, destacando o Taverna como o principal em utilização entre os membros do repositório, conforme destaca a Figura 4.24. Apesar de ser um número considerável de *workflows*, a quantidade é inferior a metade do total disponíveis no repositório, o que pode enviesar o estudo.

Com relação ao histórico de versão dos *workflows*, observou-se que entre os 1571 *workflows* analisados, apenas 29% possuem dados de versionamento, o que reitera a falta de informações sobre o ciclo de vida do *workflow*, dificultando a sua reutilização. Esta comparação é apresentada na Figura 4.25.

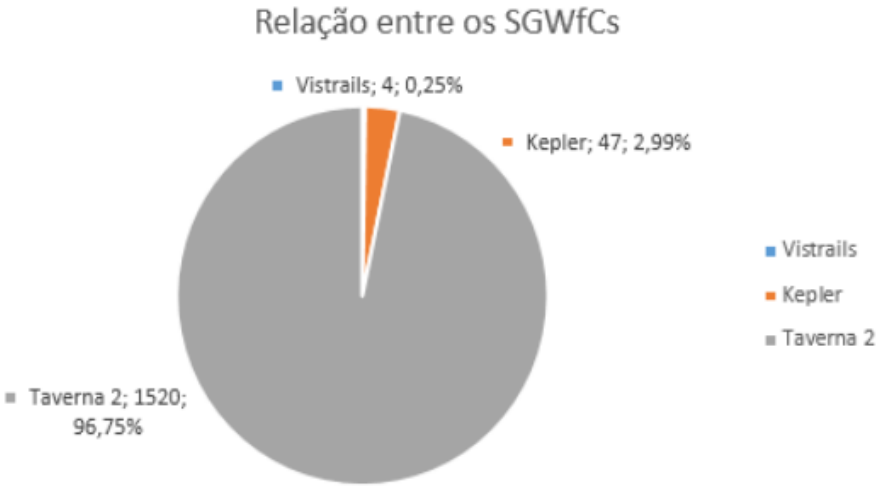


Figura 4.24: Relações de *workflows* entre os SGWfC Taverna, Kepler e Vistrails.

Figura 4.25: Comparação dos *workflows* que apresentam versionamento com os que não possuem.

Dos *workflows* que possuem informações de histórico, os mesmos são compostos por 7,22 tarefas em média. Dentre os *workflows* do Taverna, que representam 96,75% do total analisado, nota-se que a maior parte possui mais de 10 tarefas, entretanto, o número de *workflows* com tarefas em comum eram dispersos conforme pode ser visto na Tabela 4.1, um detalhe a ser observado é que 59 *workflows* não estavam disponíveis para *download* e análise.

Tabela 4.1: Total de *workflows* com o mesmo número de tarefas

Nº de Tarefas	Total de *workflows*	Porcentagem (%)
0	8	0,55
1	154	10,60
2	191	13,15
3	155	10,67
4	115	7,92
5	100	6,88
6	83	5,71
7	49	3,37
8	60	4,13
9	39	2,68
10	37	2,61
> 10	470	31,68

Foi realizada ainda uma pesquisa a fim de identificar os pesquisadores que tiveram o maior número de *workflows* disponibilizados. Esta análise foi feita com o objetivo de identificar o nível de similaridade dos mesmos, podendo indicar uma abordagem descendente na construção de novos *workflows*. A base do myExperiment conta atualmente com 10.322 membros. Destes, 236 pesquisadores estão relacionados aos *workflows* analisados. A lista dos 10 pesquisadores que mais contribuíram pode ser vista na Figura 4.26.

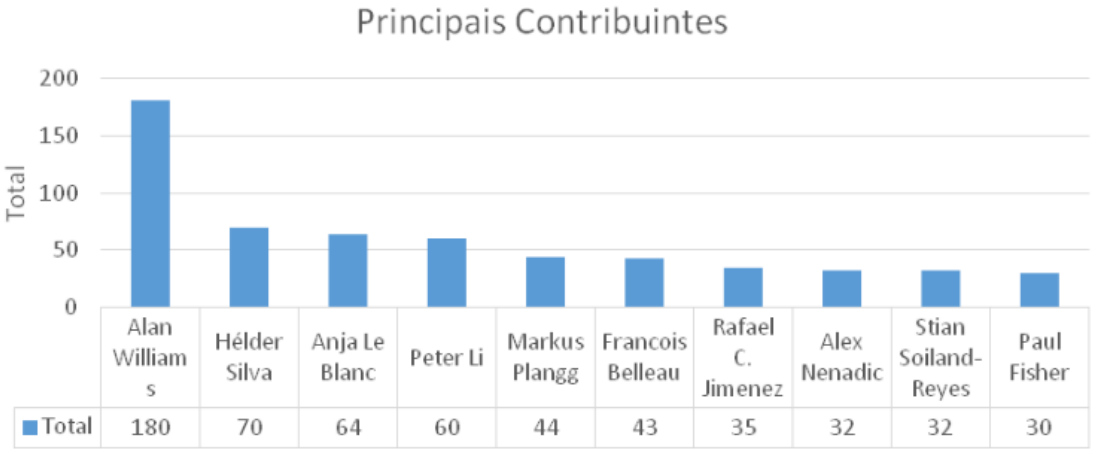


Figura 4.26: Lista dos principais contribuintes de *workflows* do myExperiment.

Por fim, a última análise realizada foi para a verificação das tarefas mais utilizadas entre os *workflows* estudados. Essa análise é útil por apresentar a quantidade de *workflows* que são afetados caso uma tarefa seja modificada. A lista das 10 tarefas mais utilizadas pode ser vista na Figura 4.27.

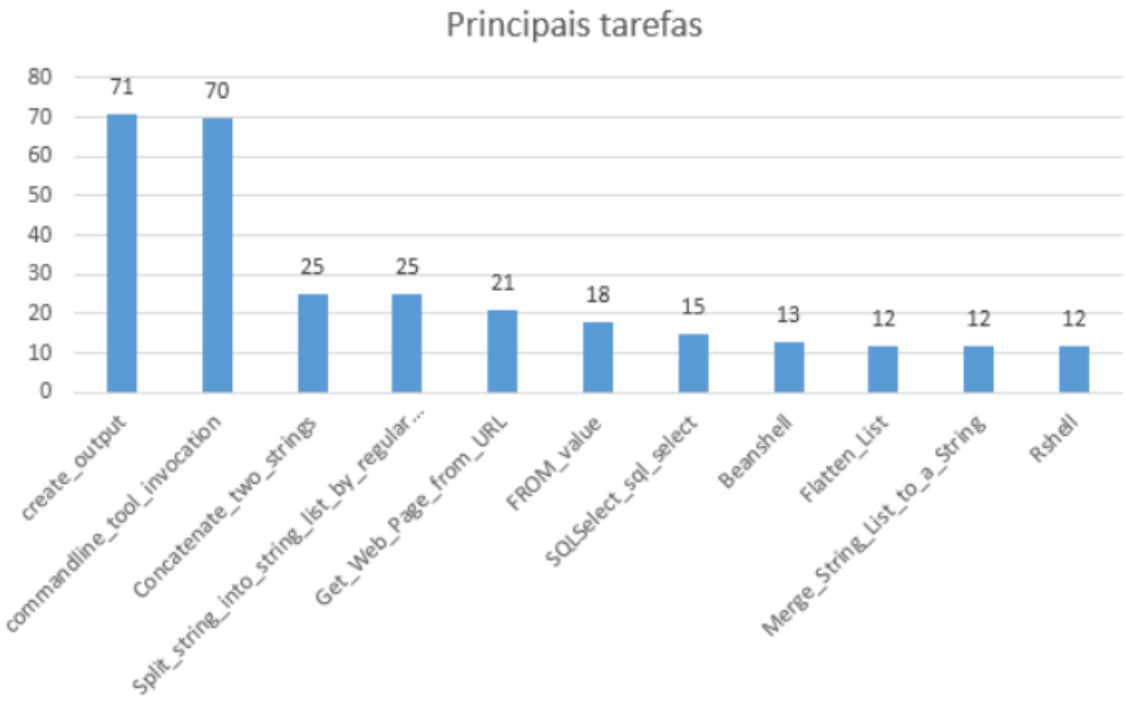


Figura 4.27: Lista das tarefas mais utilizadas nos *workflows*.

No repositório do myExperiment não existem informações sobre a procedência dos *workflows* disponibilizados e aos experimentos em que foram aplicados, dificultando a sua reutilização. Do ponto de vista da manutenção e evolução, tais informações se tornam essenciais para entender como os mesmo foram mantidos e evoluídos ao longo de um ciclo de experimentação.

4.5.2 DISCUSSÕES

Com base nas análises realizadas nos repositórios do myExperiment e do CrowLabs, identificou-se que os mesmos possuem deficiências considerando as informações de evolução e manutenção de *workflows*, não apresentando recursos para seu tratamento, que consideramos ser de grande valia para o reuso dos mesmos. O myExperiment que possui informações sobre o versionamento dos *workflows*, não permite o conhecimento detalhado sobre:

i. como o mesmo foi mantido e evoluído ao longo de um ciclo de experimentação;

ii. se é derivado de outro *workflow*;

iii. o que difere seu funcionamento do *workflow* que serviu de base;

iv. quais as tarefas que são utilizadas e o histórico de cada versão;

v. as informações sobre os serviços externos que o mesmo faz uso;

vi. os parâmetros para utilização do mesmo;

vii. a quais experimentos o mesmo foi aplicado.

Essas informações podem ser geradas a partir de dados de proveniência, que alinhados ao uso da E-SECO ProVersion, podem gerar informações que contribuam para a gerência de configuração dos experimentos científicos.

O reuso de *workflows* pode reduzir o trabalho do pesquisador no contexto de um experimento, mas para que essa reutilização seja possível, o *workflow* deve possuir informações sobre seu histórico de modo a contribuir para que o pesquisador saiba como pode reaproveita-lo. Tanto a manutenção como a evolução do *workflow* deve ser tratada como uma etapa constante dentro do ciclo de experimentação e depende de que os repositórios tenham dados histórico para permitir seu reuso.

Assim, a proposta da E-SECO ProVersion é fornecer um meta-repositório de informações sobre os *workflows*, acoplando-se aos repositórios já existentes, fornecendo informações adicionais a partir do uso de ontologias, de modo a contribuir para a formação de uma base de conhecimento sobre os mesmo, contribuindo para a gerencia de configuração dos experimentos.

Assim, para auxiliar a gerência de configuração de experimentos científicos com a finalidade de reutilização ou replicação, algumas informações devem ser registradas junto ao experimento. Nessa perspectiva, um meta-repositório para gerência de configurações de experimentos científicos deve ser criado, atendendo tanto aos requisitos do experimento quanto do *workflow*, visto que o mesmo possui influência direta no experimento. As seguintes características devem estar presentes no repositório:

- Gestão dos *workflows*: armazenar informações sobre quais *workflows* seriam afetados durante a modificação de uma atividade, as tarefas que compõem cada *workflow*, o registro do fluxo entre as tarefas, como os mesmos foram mantidos e evoluídos,

informações sobre versionamento, busca por *workflows* similares, informações de proveniência sobre sua criação e aplicação;

- Gestão dos experimentos: armazenar informações sobre a criação, condução e resultados obtidos por cada experimento, registrar quais *workflows* estão ligados a quais experimentos, gerenciar os grupos de pesquisa ou pesquisador vinculado a cada experimento, registrar as informações sobre as manutenções e evoluções, bem como as versões existentes de um experimento;

- Sistema colaborativo: apoiar a gestão dos *workflows* que compõem o experimento sobre o uso de laboratórios colaborativos, acompanhamento do ciclo de experimentação registrando todas as informações sobre os *workflows* e experimentos para compartilhamento do conhecimentos, permitir a criação de laboratórios colaborativos e a formação de grupos de pesquisa trabalhando simultaneamente;

- Armazenamento dos dados: registrar os parâmetros e informações do experimento, capturando todos os passos executados ao longo dos ciclos de experimentação, bem como os dados consumidos, processados e produzidos em cada *workflow*, garantir escalabilidade e flexibilidade do repositório, seguir como base um modelo de proveniência padronizado;

- Repositório: permitir o acesso público, armazenando e registrando as informações dos *workflows*, dos experimentos e dos pesquisadores de modo a compartilhar o conhecimento com a comunidade cientifica;

Essas características compõem o grupo de requisitos básicos que devem existir em um repositório de gerência de configuração de experimentos científicos. As características descritas para gestão de experimentos e de sistema colaborativo já existiam no E-SECO, as demais características foram expandidas na E-SECO ProVersion e podem ser utilizadas pelos pesquisadores no âmbito de seus experimentos.

4.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou detalhes da plataforma de ecossistemas E-SECO, discutiu algumas características de um ecossistema de software científico e o ciclo de vida de um

experimento. Discutiu-se também aspectos da arquitetura E-SECO ProVersion, apresentando o novo ciclo de vida do experimento a ser abordado, detalhes do esquema do banco de dados e da extensão da ontologia PROV-O denominada PROV-OEXT. Foram também detalhados os módulos que passam a compor a E-SECO ProVersion, bem como suas características, modo de operação e finalidade de uso. Também foi apresentado uma breve análise dos repositórios de *workflows* existentes e discutidos os requisitos mínimos para um repositório de apoio à manutenção e evolução de experimentos científicos.

No capítulo seguinte é apresentada a avaliação da abordagem E-SECO ProVersion, envolvendo uma prova de conceito de uso da arquitetura.

5 DIRETRIZES DE UTILIZAÇÃO

Conforme ressaltado nos capítulos anteriores, a arquitetura E-SECO ProVersion possui como objetivo auxiliar os pesquisadores na gerência de configuração de experimentos científicos e seus *workflows* relacionados, através do uso de um modelo de proveniência e de ontologias. No entanto, parte primordial para o aprimoramento dos resultados é a disponibilidade de dados históricos acerca dos experimentos e *workflows* vinculados.

O uso de ontologias e mecanismos de inferência auxiliam na derivação de conhecimento implícito para o aprimoramento da gestão de experimentos e de *workflows*, mas informações importantes também podem ser descobertas a partir dos dados históricos. Considerando esse contexto de dados históricos, um dos problemas encontrados para análise de *workflows* científicos e seus experimentos é justamente a falta de dados que permitam realizar análises relativas à evolução e manutenção e, com isso, propor melhorias utilizando esses dados, bem como auxiliar o pesquisador na tomada de decisão. Apesar de existirem repositórios com dados de *workflows* como o CrowDLabs (CALLAHAN *et al.*, 2006) e o myExperiment (GOBLE *et al.*, 2010) , esses são incompletos, com informações inconsistentes e, na maioria das vezes, não muito organizados do ponto de vista de metadados. Além disso, buscando analisar experimentos científicos, não foram encontrados repositórios públicos com dados de pesquisas para avaliação.

Assim, considerando o uso de ontologias e mecanismos de inferência e a falta de dados históricos relacionados a *workflows* e experimentos científicos, apresentamos neste capítulo algumas diretrizes de utilização da arquitetura proposta com objetivo de realizar uma avaliação preliminar [1] considerando dois contextos: i) a pouca disponibilidade de dados históricos relacionados a *workflows* e experimentos científicos; ii) o uso de ontologias e mecanismos de inferência. Considerando i), foi realizado um estudo com *workflows* extraídos do myExperiment, com o objetivo de demonstrar o uso da arquitetura para extrair informações sobre manutenção e evolução nos *workflows*, bem como a identificação de *workflows* similares e, por fim, como resultado de i) foi realizado ii), uma prova de conceito, onde *workflows* selecionados do repositório myExperiment foram utilizados para

[1] Não é objetivo deste capítulo realizar uma avaliação criteriosa da arquitetura, mas sim obter indícios relacionados à sua utilização.

demonstrar o uso da arquitetura, quanto a captura e inferência das informações.

Além dos contextos descritos acima, como o objetivo de demonstrar as principais funcionalidades do E-SECO ProVersion e familiarizar o leitor com a utilização da abordagem, foi elaborado, na seção seguinte, 5.1, um roteiro de utilização da arquitetura.

5.1 ROTEIRO DE UTILIZAÇÃO DA ARQUITETURA E-SECO PROVERSION

Esta seção apresenta um roteiro de utilização da arquitetura E-SECO ProVersion, apresentando os passos que devem ser executados pelo pesquisador para sua utilização e quais os recursos disponíveis. Para esta apresentação, será seguido o fluxograma apresentado na Figura 5.1.

Para apresentar as funcionalidades da arquitetura, utilizamos dois *workflows* denominados "BuscaGENE" e "GeneExtraction", que são *workflows* científicos simples mas permitem a apresentação das principais funcionalidades da arquitetura. Foram preparadas 3 versões do primeiro *workflow*, representando o *workflow* inicial e uma simulação da sua evolução ao longo do tempo e 2 versões do segundo para busca de *workflow* similares, de forma a apresentar como a arquitetura E-SECO faz o controle dos dados e os exibe ao pesquisador.

A arquitetura E-SECO trata a evolução dos *workflow* de duas maneiras. A primeira de forma vertical, onde são analisadas todas as versões de um mesmo *workflow*. Na forma vertical, a arquitetura trata versões que surgiram por manutenções corretivas ou reengenharia (otimização, por exemplo) em que não se tem grandes diferenças entre as versões, mas seu controle é importante no contexto da gerência de configuração. A segunda forma é a evolução horizontal ou linha do tempo. Nessa segunda forma, são avaliados i) como os *workflows* evoluíram ao longo do tempo, acrescentando recursos que antes não faziam parte de seu escopo inicial e ii) como outros *workflows* foram desenvolvidos com base nesse. Essa última forma caracteriza o uso de uma abordagem descendente na construção dos *workflows* e poderia ser denominada como família de *workflows*, visto que todos apresentam características em comum e uma mesma linha de derivação.

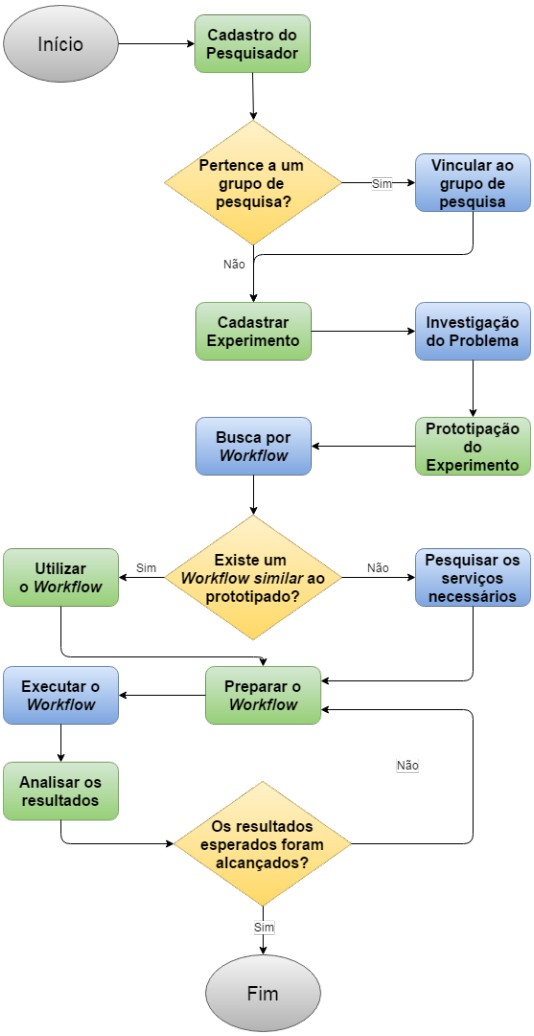


Figura 5.1: Fluxograma de apresentação da arquitetura E-SECO ProVersion.

Para detalhar um roteiro de utilização da arquitetura, apresenta-se na Figura 5.1, um fluxograma de uso da arquitetura que se inicia pela tela de cadastro de pesquisador, Figura 5.2. Através desse cadastro, o pesquisador pode fazer parte de um grupo de pesquisa existente ou trabalhar de forma autônoma.

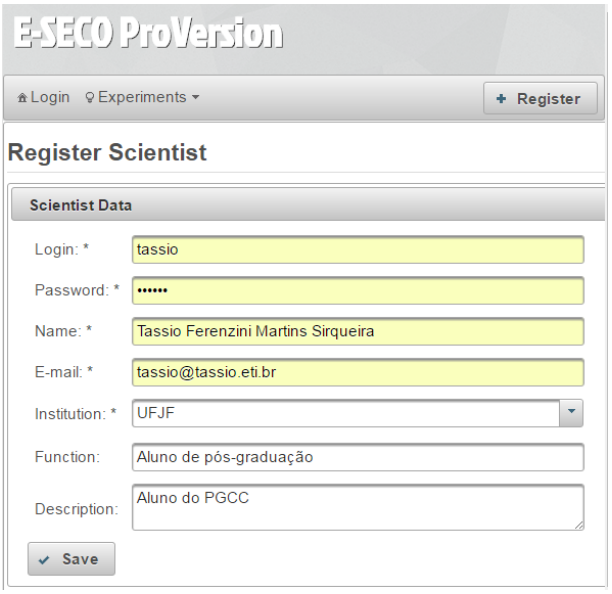


Figura 5.2: Cadastro dos Pesquisadores na E-SECO ProVersion.

Como a proveniência é um dos focos deste trabalho, conhecer o pesquisador ou grupo de pesquisa relacionado a um experimento é importante, visto que na publicação de resultados da pesquisa, esses são vinculados a um pesquisador responsável, o que traz confiabilidade e permite o rastreio das origens da pesquisa. No caso de grupos de pesquisa, o criador do grupo de torna o responsável pelo mesmo e assim pode gerenciar quais outros pesquisadores fazem parte do grupo, conforme Figura 5.3. Essa gestão de grupos de pesquisa para a E-SECO é primordial, visto que todos os dados de um experimento devem estar disponíveis a todos os membros dos grupos de pesquisa, assim como os resultados de execução e a evolução da pesquisa. Essa característica é importante principalmente para o uso em laboratórios colaborativos, onde os pesquisadores encontram-se dispersos geograficamente e é importante que todos tenham acesso ao andamento da pesquisa, os passos já executados e as próximas tarefas a serem realizadas.

Continuando o roteiro e considerando especificamente a gerência de experimentos, essa inicia-se com a criação das atividades que compõem o experimento, de forma a definir o que esse deve realizar. Para cada experimento, é definido um nome com informações de grupo de pesquisa ou pesquisador associado, as datas de início e fim do experimento, sua descrição e versão, conforme Figura 5.4. É importante destacar que um mesmo experimento pode conter diferentes versões, pois a cada mudança em sua configuração é gerada uma nova versão.

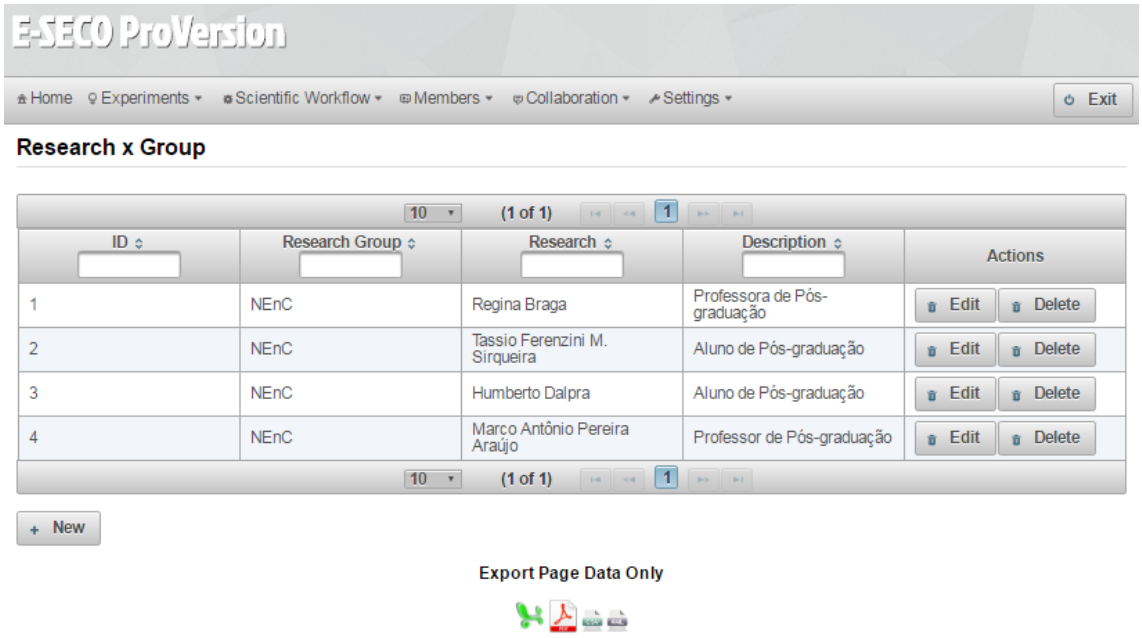


Figura 5.3: Gestão dos grupos de pesquisa na E-SECO ProVersion.

O controle dessas alterações faz parte da gerência de configuração do experimento e registra todos os acontecimentos ao longo do ciclo de experimentação, de forma otimizada, ou seja, sem gerar grande volume de dados, como destacado no capítulo 4.

Esse recurso da E-SECO ProVersion permite ao pesquisador conhecer o andamento da pesquisa, comparar resultados de ciclos de experimentações diferentes, além de disponibilizar todo o conhecimento do estudo com todos os pesquisadores envolvidos.

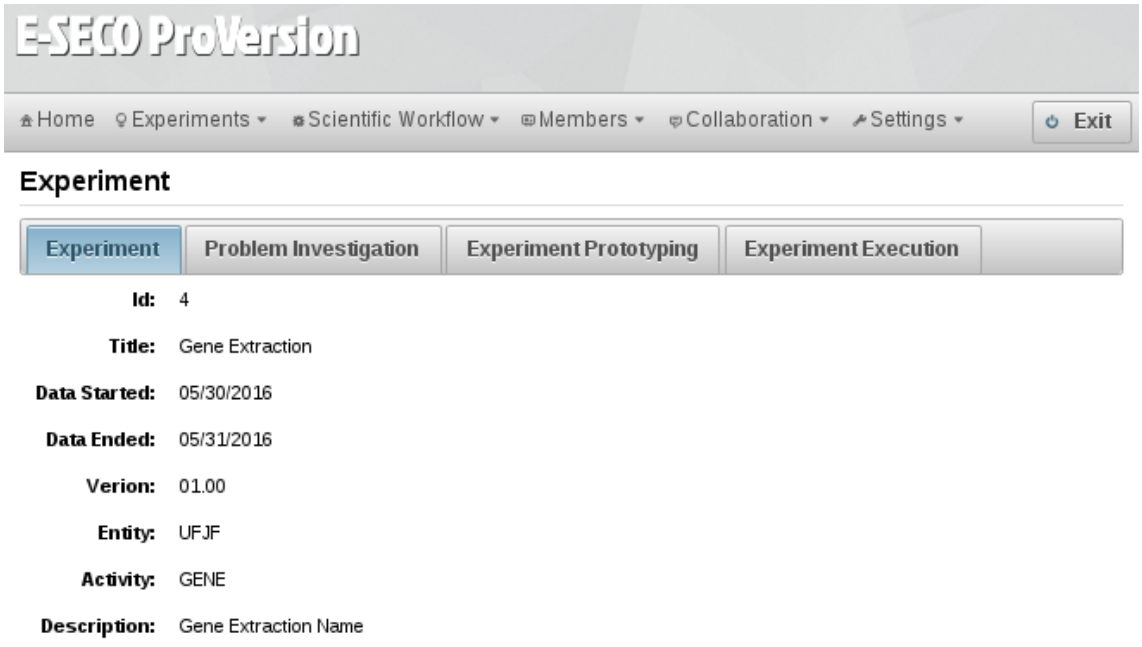


Figura 5.4: Gestão do experimento na E-SECO ProVersion.

Considerando ainda o experimento, a E-SECO ProVersion permite, na tela de investigação do problema, Figura 5.5, verificar os experimentos relacionados com o selecionado, e consultar revisões bibliográficas que foram realizadas no Parsifal [2] ou Mendeley [3] relacionadas ao mesmo. Esse recurso ressalta a característica de que todo o conhecimento sobre o experimento possa ser acessado a partir de um único ponto, mesmo considerando a característica distribuída que o experimento possa ter.

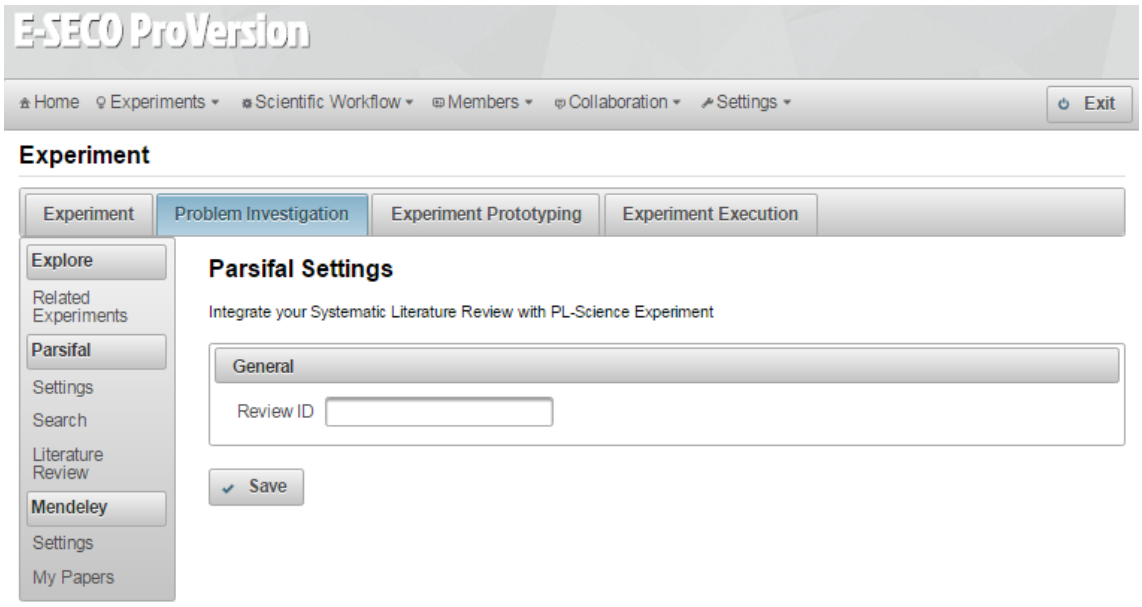


Figura 5.5: Gestão dos experimentos relacionados e revisões bibliográficas.

Avançando no fluxograma de utilização, já na tela de prototipação do experimento, Figura 5.6, a E-SECO permite ao pesquisador compor o número de estágios do(s) *workflow*(s) que será(ão) utilizado(s) no experimento e pesquisar os serviços disponíveis na plataforma E-SECO ou em um catálogo de serviços externos, como no BioCatalogue (BHAGAT *et al.*, 2010) . Outro recurso importante é permitir ao pesquisador verificar se existe algum *workflow* similar ao que se deseja, evitando o retrabalho para a construção de um novo ou permitindo-se utilizar algum *workflow* próximo ao desejado.

Assim, ao invés do pesquisador ter que criar um novo *workflow*, pode-se utilizar como base algum já existente e através do uso de abordagem descendente, compor, através de modificações no *workflow* base, o novo *workflow*.

[2] http://parsif.al/
[3] https://www.mendeley.com/

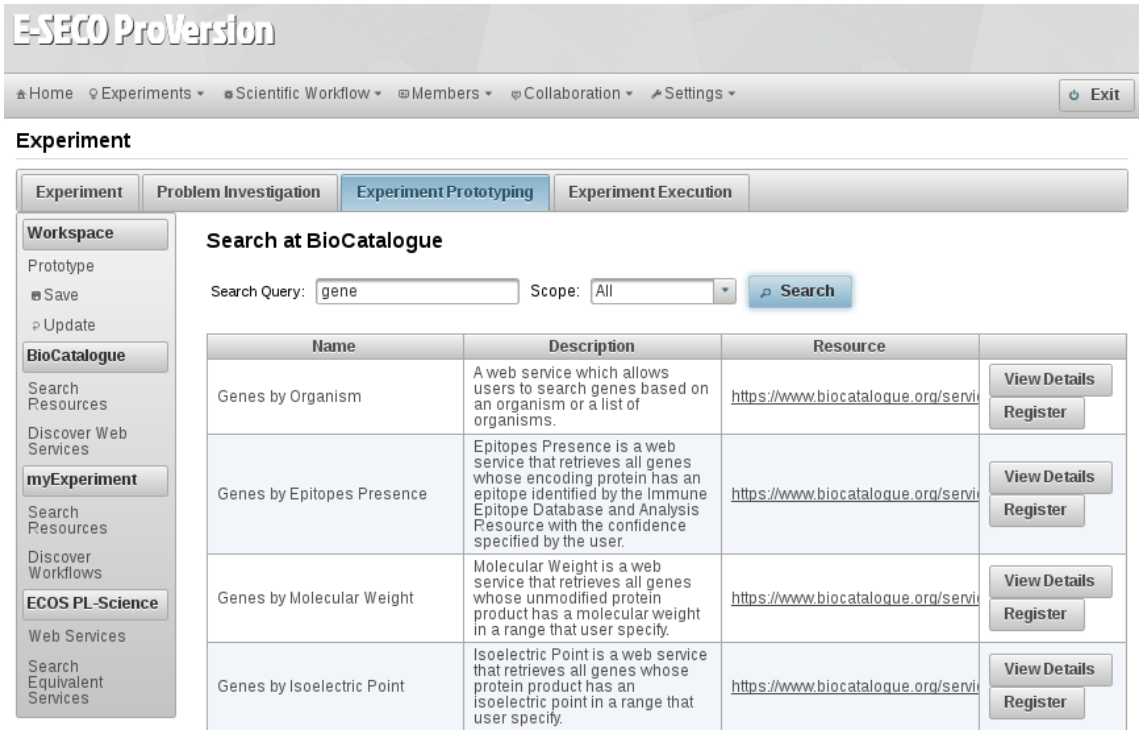


Figura 5.6: Prototipação do experimento na E-SECO ProVersion.

A Figura 5.7 apresenta a consulta de *workflows* que estão relacionados ao experimento, consulta dos recursos (serviços e tarefas) que foram utilizados e o andamento dos mesmos.

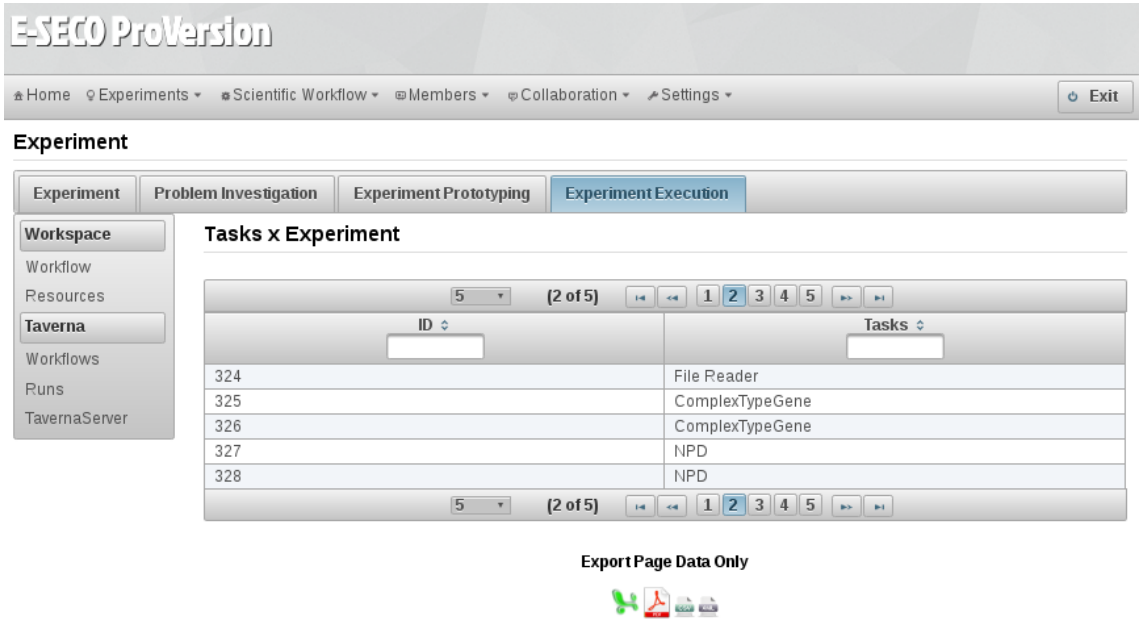


Figura 5.7: Andamento do experimento na E-SECO ProVersion.

Seguindo o roteiro de utilização, considerando agora a gerência de *workflows*, a arquitetura E-SECO ProVersion permite atribuir informações como nome, SGWfC utilizado, descrição, número de estágios e local, sendo que o *workflow* pode ser armazenado em um

repositório local na plataforma E-SECO ou indicado por um link o local onde encontra-se disponível, bem como as informações de versão e data da versão, conforme Figura 5.8.

Para apresentar os recursos da arquitetura será utilizado o *workflow* "BuscaGENE", inicialmente na versão 1.00.00, conforme pode ser visto na Figura 5.9.

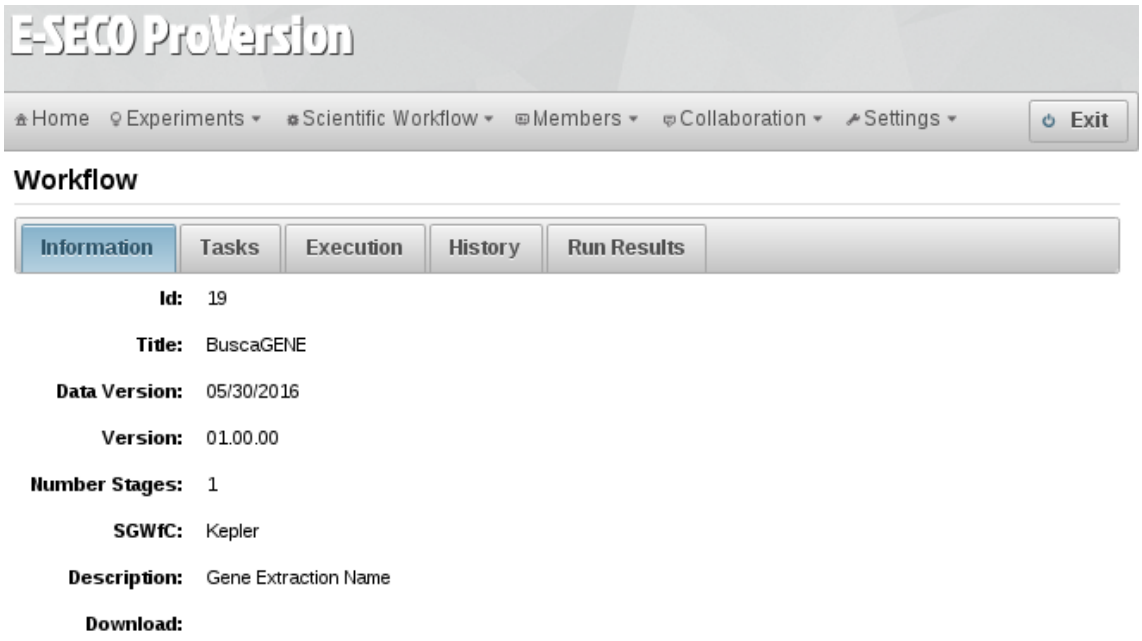


Figura 5.8: Gestão dos *workflows* na E-SECO ProVersion.

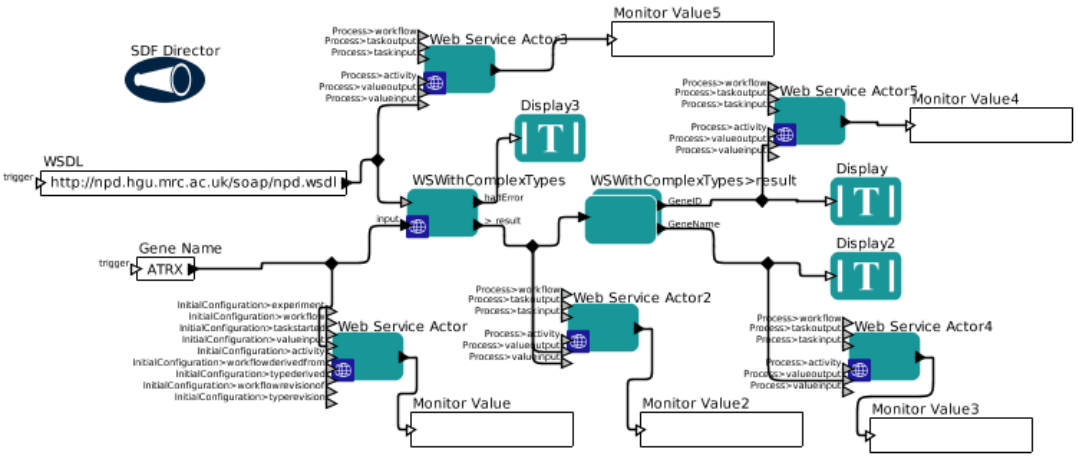


Figura 5.9: *Workflow* BuscaGENE versão 1.00.00.

O *workflow* BuscaGENE foi desenvolvido para buscar as informações de Gene em um serviço web, retornando as informações do nome e identificador. Para que os dados de execução sejam capturados pela E-SECO ProVersion, é necessário realizar a preparação do mesmo, conforme apresentado no fluxograma da Figura 5.1. Dessa forma, *workflow* BuscaGENE foi instrumentalizado com um serviço web disponibilizado pela E-SECO ProVersion. Essa instrumentalização é feita com duas opções. A primeira é o "InitialConfiguration", que é responsável por capturar as informações iniciais do *workflow*, conforme Figura 5.10.

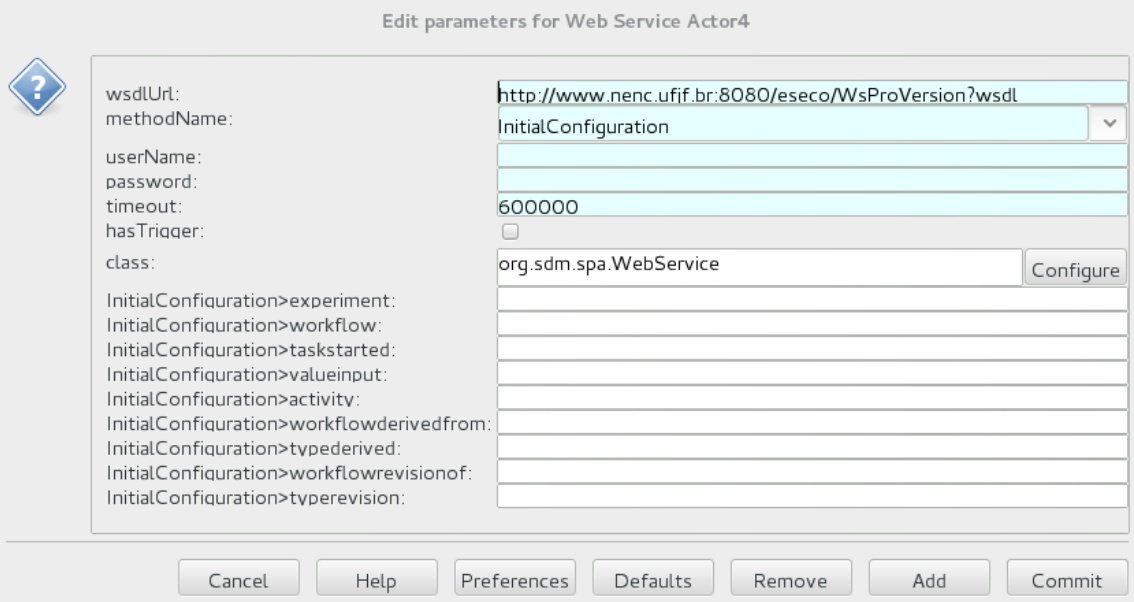


Figura 5.10: Serviço "InitialConfiguration" da E-SECO ProVersion.

A segunda opção é o "Progress", que realiza a captura de proveniência de todos os componentes, serviços ou tarefas do *workflow* ao longo de sua execução. Sua tela de configuração pode ser vista na Figura 5.11.

Ao executar o *workflow* com sucesso, tendo todos os dados sido capturados, a E-SECO ProVersion exibe como retorno a mensagem "NO ERRORS". Para demonstrar como a arquitetura trata a manutenção e a evolução dos *workflows*, foi realizada uma manutenção no *workflow* BuscaGENE versão 1.00.00 para que o mesmo passasse a ler um arquivo de texto contendo um conjunto de genes e exibisse os resultados na tela em formato texto. Essa nova versão foi especificada como 1.01.00 e pode ser vista na Figura 5.12.

Ao executar o *workflow*, a E-SECO ProVersion possibilita a identificação de vínculo com a versão anterior do *workflow* de duas maneiras: através da ontologia por meio

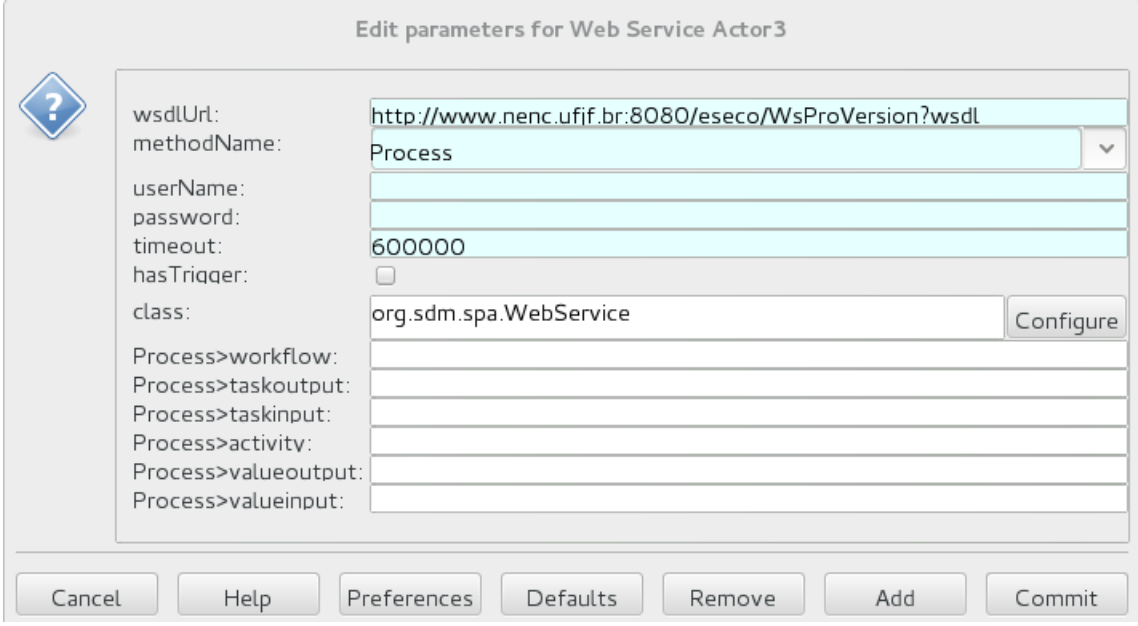


Figura 5.11: Serviço "Progress" da E-SECO ProVersion.

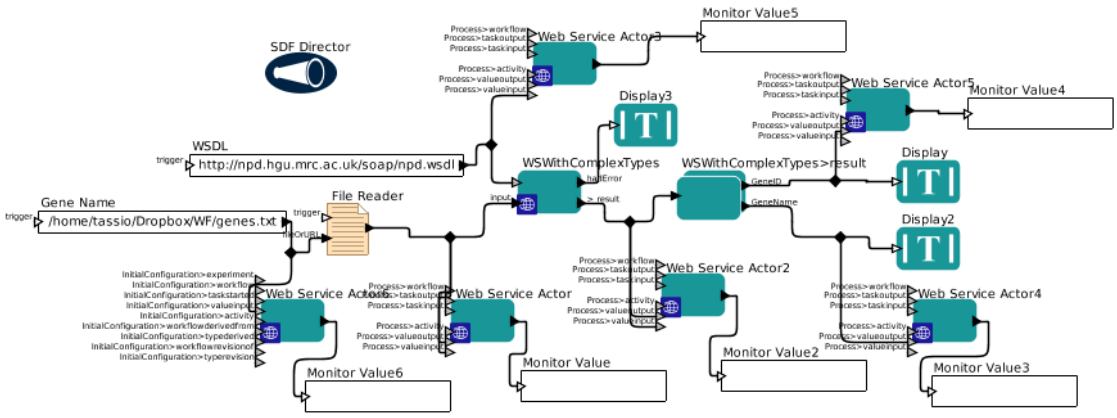


Figura 5.12: *Workflow* BuscaGENE versão 1.01.00.

de inferência utilizando o raciocinador Pellet, ou o pesquisador informa explicitamente para a arquitetura. O pesquisador pode informar esses dados na configuração do serviço "InitialConfiguration", onde pode se informar de qual *workflow* o novo foi revisado e o tipo de manutenção. Caso contrário, a ontologia fica responsável por essa inferência, de maneira automática, sem intervenção do pesquisador.

Assim, quando o pesquisador consultar as informações do *workflow* "BuscaGENE" versão 1.01.00, será apresentado que o mesmo é uma manutenção corretiva do *workflow* BuscaGENE versão 1.00.00, conforme Figura 5.13. Todos os dados capturados na execução do *workflow* e os dados informados pelo pesquisador são repassados para a ontologia para inferência. Com isso, todas as consultas são realizadas na ontologia PROV-OEXT,

de forma a extrair conhecimento implícito.

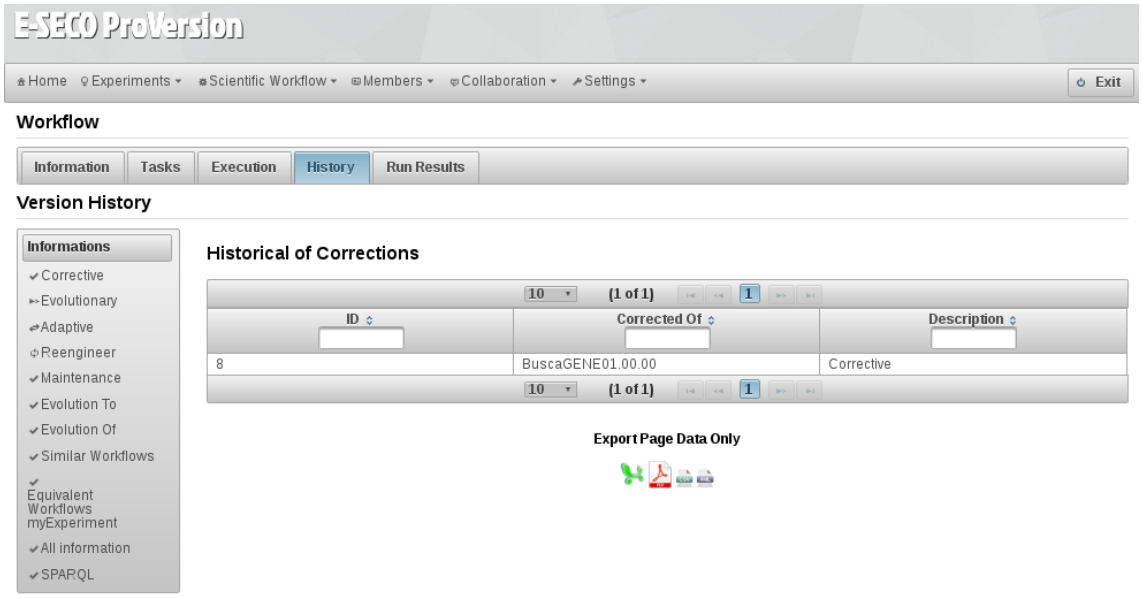


Figura 5.13: Histórico de correção do *Workflow* BuscaGENE.

Quando o objetivo é registrar no histórico do *workflow* que o mesmo sofreu uma evolução, o pesquisador pode fazer isso também por meio do serviço "InitialConfiguration", indicando no parâmetro "workflowderivedfrom" de qual esse foi evoluído e qual o motivo da evolução, ou então essa evolução pode ser identificada automaticamente, utilizando a ontologia PROV-OEXT.

Para representar esta situação foi criado o *workflow* "BuscaGENE" versão 01.01.01. O nome e a indicação da versão é de responsabilidade do pesquisador, onde mesmo que não exista similaridade de nomes ou proximidades de versão, a ontologia possibilita as inferências por meio das regras e restrições existentes.

No exemplo apresentado, foi adotada uma convenção em que os dois primeiros dígitos representam mudanças significativas entre as versões, os dois segundos dígitos indicando mudanças menores como, por exemplo, adição ou modificação de uma tarefa ou serviço na composição do *workflow* que impacte na saída dos dados e os dois últimos dígitos mudanças que não alterem os resultados produzidos pelo *workflow*. O *workflow* "BuscaGENE" versão 1.01.01 apresenta a mesma composição de tarefas do anterior, mas os resultados são exibidos ao pesquisador em formato XML, conforme Figura 5.14, ou seja, uma manutenção pequena onde não são alterados os dados de saída, apenas sua forma de visualização.

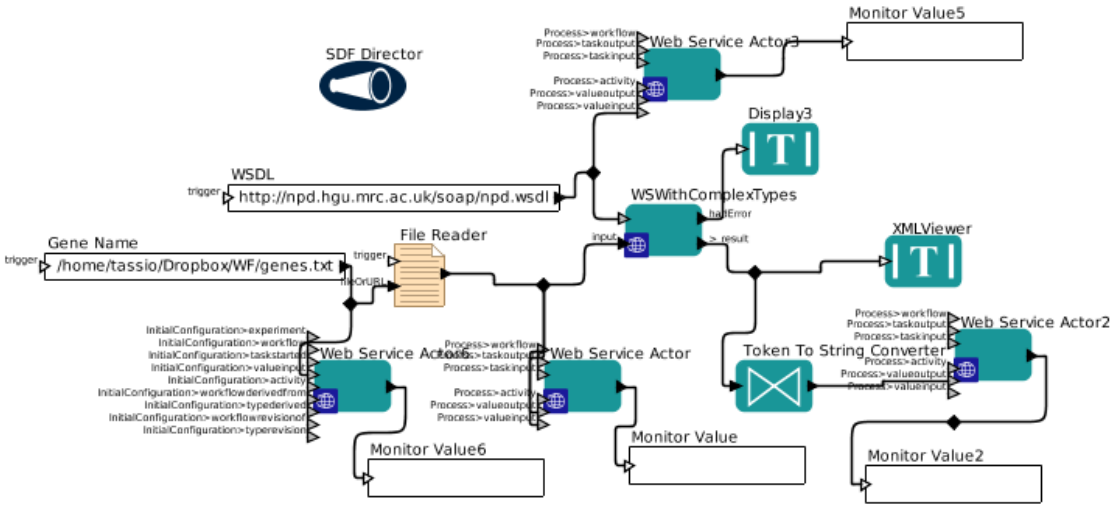


Figura 5.14: *Workflow* BuscaGENE versão 1.01.01.

Todos os dados capturados pelo serviço web ou os dados informados pelo pesquisador são repassados à ontologia onde são realizadas as inferências. De modo a apresentar informações estratégicas ao pesquisador, a E-SECO ProVersion busca todas as informações sobre os *workflows* e os experimentos na ontologia PROV-OEXT, como, por exemplo, é apresentado na Figura 5.15, referentes às informações sobre o *workflow* "BuscaGENE" versão 01.01.01.

A E-SECO ProVersion permite consultar com base no *workflow* selecionado, de qual este foi evoluído ou para qual este evoluiu, conforme Figura 5.16, sendo essas informações provenientes da ontologia.

Visando demonstrar a capacidade de identificar *workflows* similares, foram construídos dois novos *workflows*, que também buscam informações de GENEs utilizando o serviço de consulta de genes do Medical Research Council [4]. Estes *workflows* foram denominados "GeneExtraction" nas versões 2.00.00 e 2.01.00, respectivamente.

O primeiro *workflow* (GeneExtraction versão 2.00.00) foi evoluído do *workflow* "BuscaGENE" versão 1.01.01, onde passou-se a armazenar os resultados dos Ids dos genes consultados em um arquivo XML. Já o segundo *workflow* (GeneExtraction versão 2.01.00) realiza as mesmas funções do seu antecessor tendo como diferença a saída dos dados, onde todas as informações são armazenadas no arquivo XML. Os dois *workflows* podem ser vistos nas figuras 5.17 e 5.18, respectivamente.

[4]http://www.mrc.ac.uk/

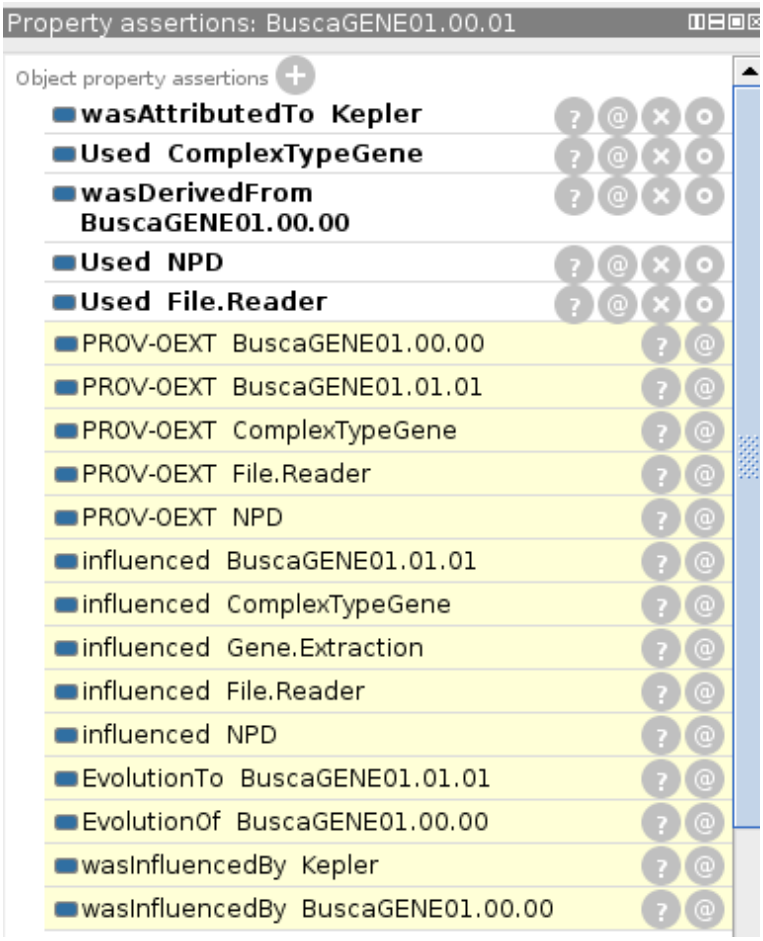


Figura 5.15: Inferências do *Workflow* BuscaGENE versão 01.01.01.

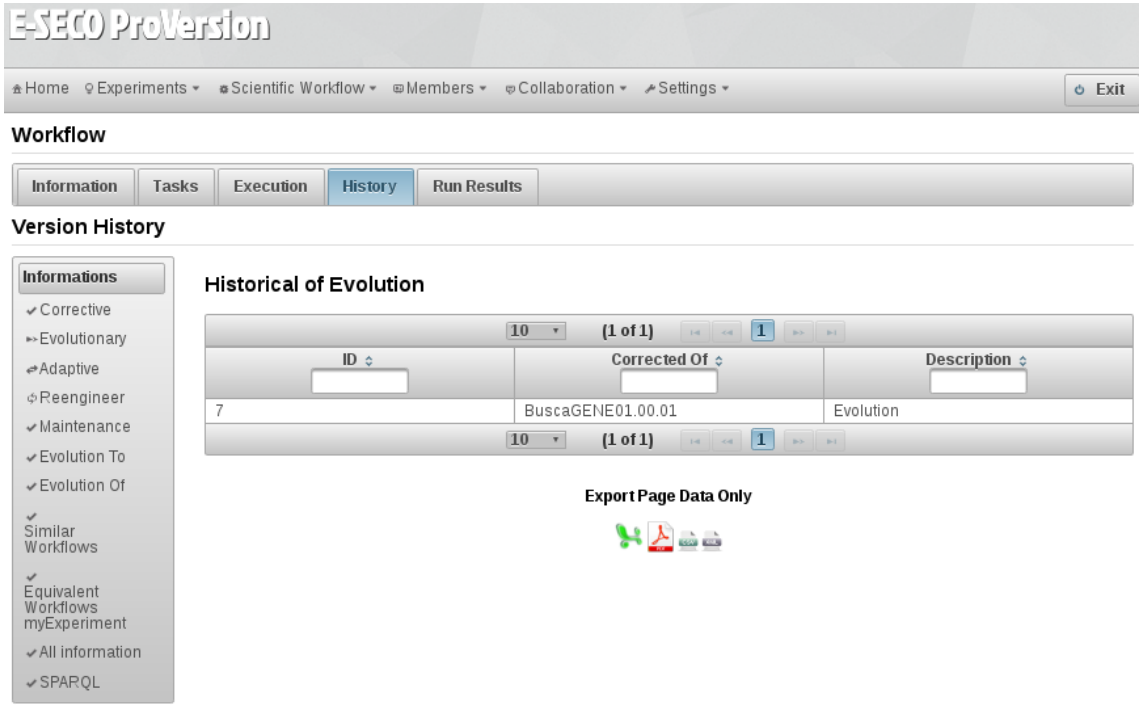


Figura 5.16: Histórico de evolução do *workflow*.

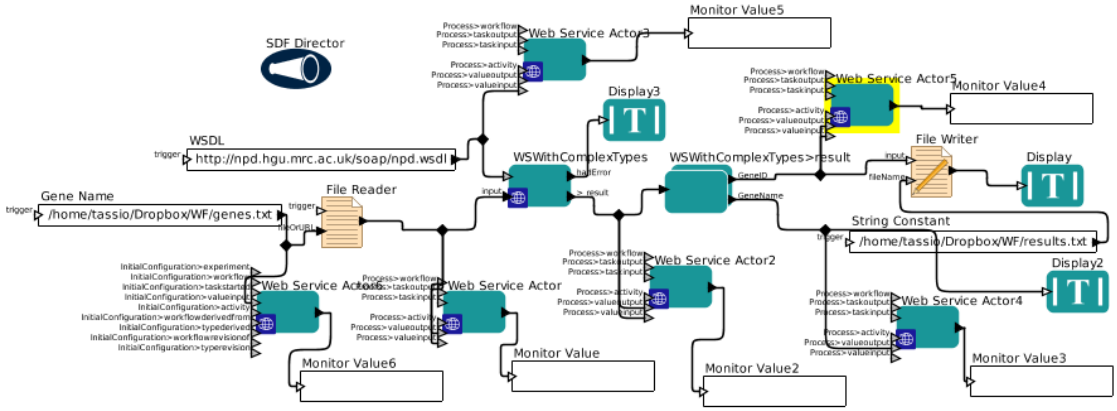


Figura 5.17: *Workflow* GeneExtraction versão 2.00.00.

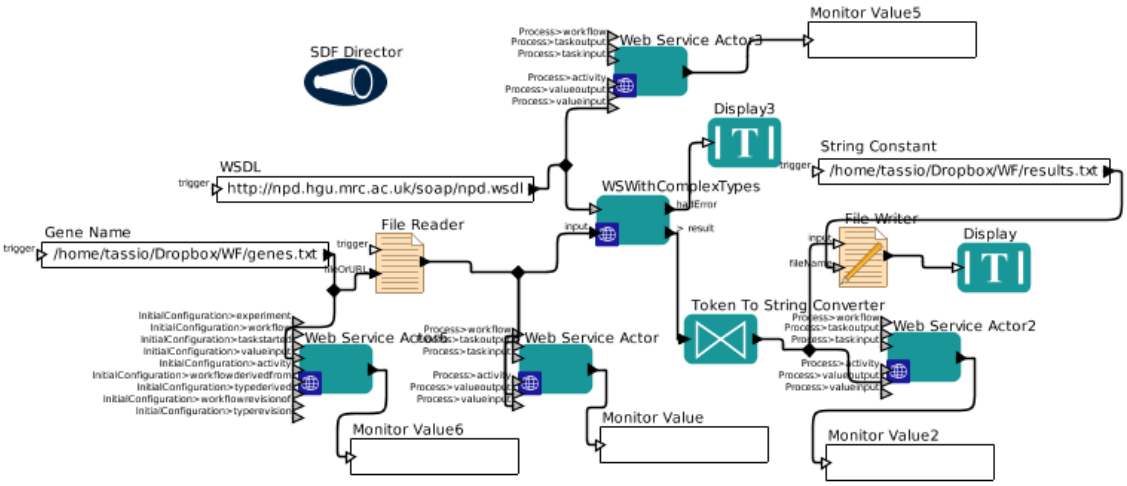


Figura 5.18: *Workflow* GeneExtraction versão 2.01.00.

Com isso, ao consultar-se os dados de execução do *workflow* "GeneExtraction" versão 2.01.00, o mesmo apresenta que esse foi uma manutenção da versão 2.00.00 e o pesquisador informou que essa nova versão foi uma reengenharia da versão anterior, conforme Figura 5.19.

Por fim, ao verificar na E-SECO ProVersion quais *workflows* possuem similaridade, a partir da seleção de qualquer uma das versões do *workflow* BuscaGENE, as demais devem aparecer, visto que todas compartilham de tarefas em comum e apresentam a mesma finalidade. O resultado dessa similaridade é exibido no histórico do *workflow*, tendo essa informação proveniente da ontologia, conforme a Figura 5.20.

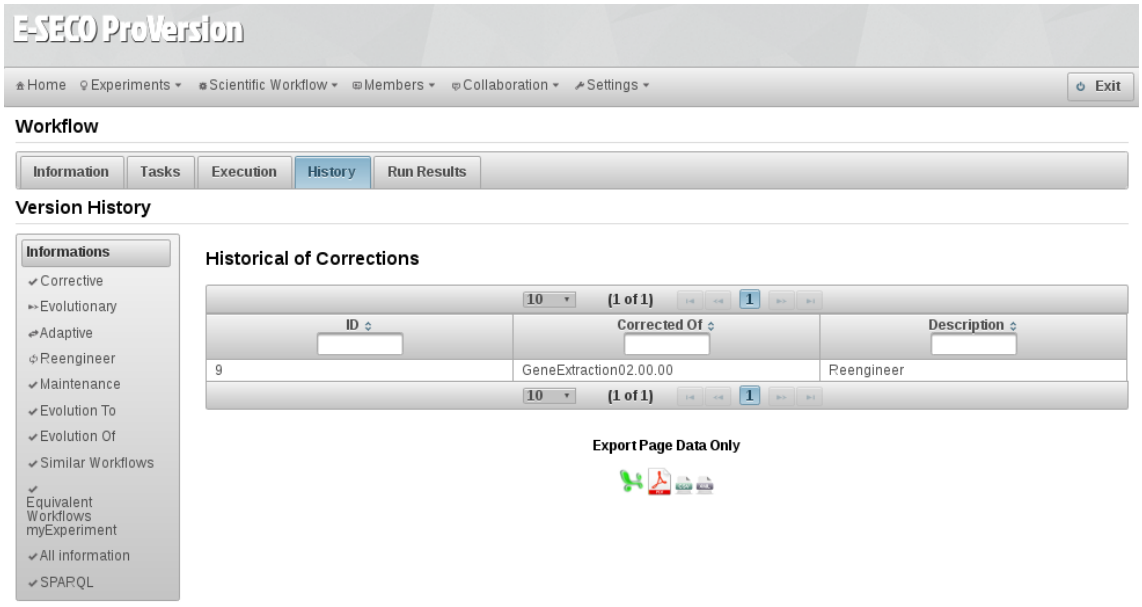


Figura 5.19: Histórico do *workflow* GeneExtraction.

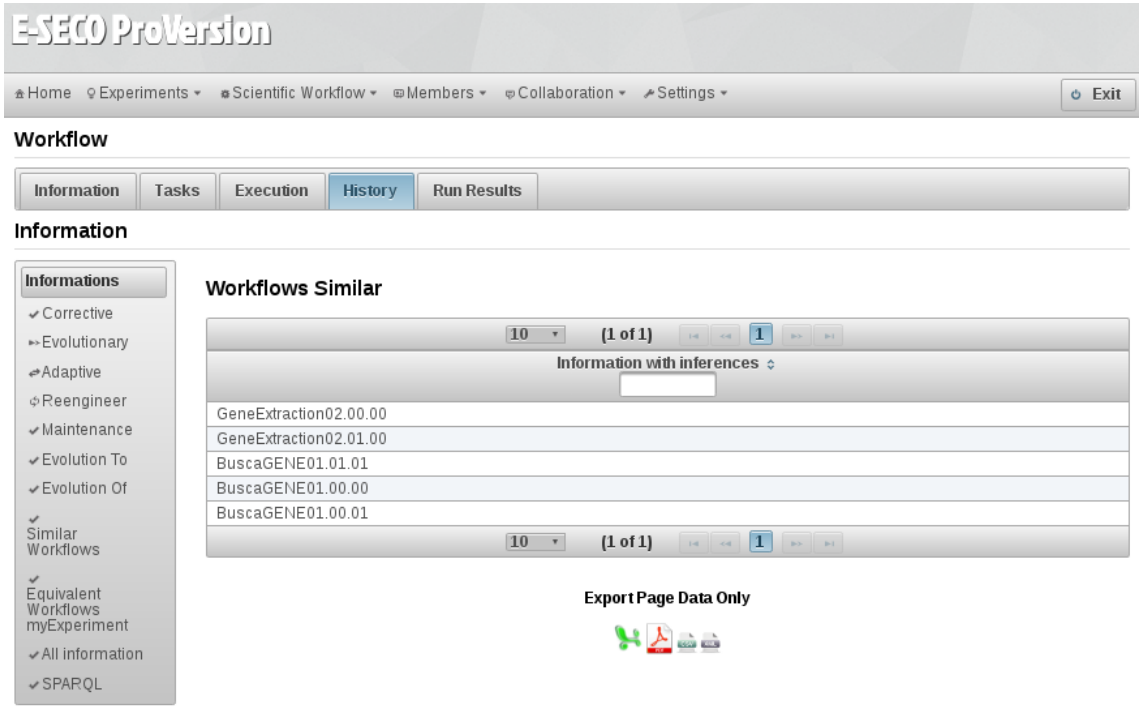


Figura 5.20: *Workflows* similares ao BuscaGENE.

Os resultados da execução de cada tarefa do *workflow*, constando sucesso ou falha, podem ser vistos nos resultados de execução, conforme apresenta a Figura 5.21. Caso tenha ocorrido alguma falha na execução do *workflow*, o mesmo deve retornar à etapa de preparação, conforme o fluxograma da Figura 5.1, que sofrerá manutenção para sua correção.

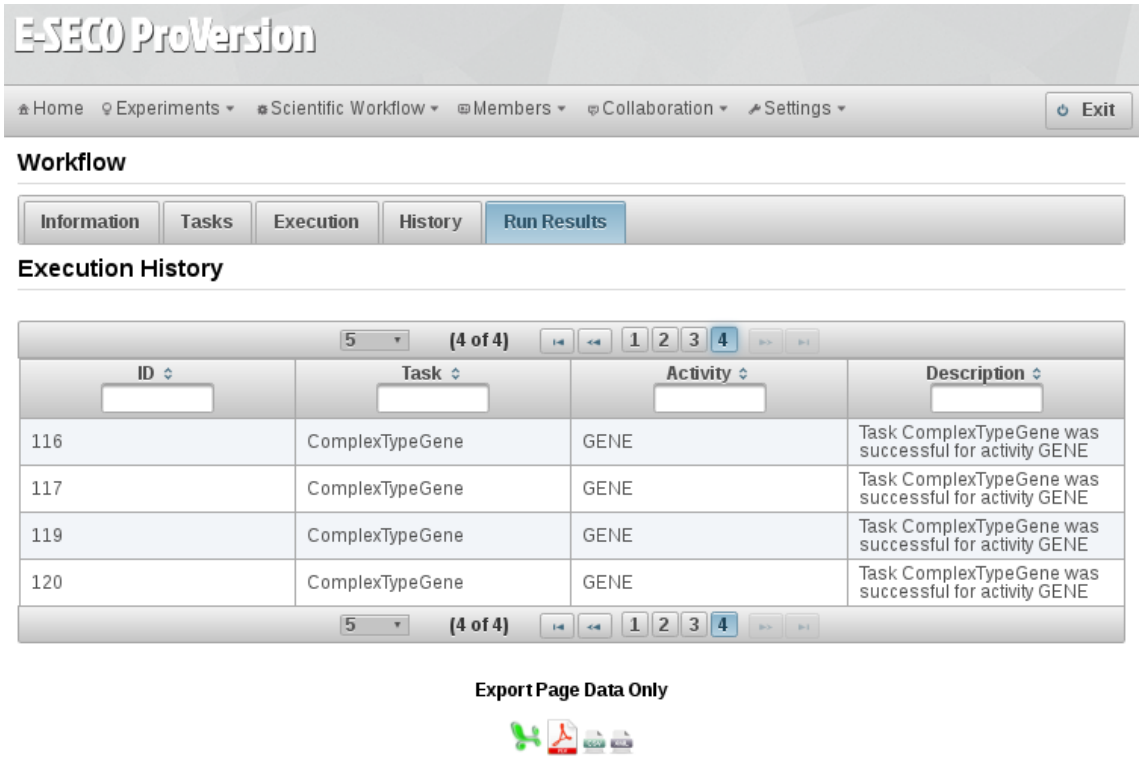


Figura 5.21: Resultado de execução do *workflow*.

Conforme dito anteriormente, o objetivo desta seção foi apresentar ao leitor um roteiro de utilização da arquitetura E-SECO ProVersion, apresentando os passos que podem ser executados pelo pesquisador e como a arquitetura pode auxiliá-lo na gestão do experimento e dos *workflows* vinculados. Na próxima seção será apresentada uma Prova de Conceitos com o objetivo de avaliar a capacidade da arquitetura em identificar manutenção e evolução de *workflows*.

5.2 PLANEJAMENTO DA PROVA DE CONCEITO

Assim, para analisar a arquitetura do E-SECO ProVersion foram especificadas i) uma análise de repositórios de *workflows* e ii) uma prova de conceito para avaliar a capacidade da arquitetura em identificar manutenção ou evolução dos *workflows*, com base na proveniência dos dados capturados pela arquitetura.

Com base nos resultados da análise realizada em i) foi realizada a prova de conceito. Uma prova de conceito é um instrumento utilizado para denominar um modelo prático que possa verificar o conceito (teoria) estabelecida em uma pesquisa (BELL *et al.*, 1994) . Em Engenharia de Software, o termo pode ser relacionado ao desenvolvimento de um

protótipo, ferramenta ou arquitetura de hardware ou software.

A prova de conceito normalmente pode explorar dois aspectos, estrutura e comportamento (MENDES, 2014) . As provas de conceito que exploram estruturas podem ser usadas para comparar ou selecionar tecnologias e as provas de conceito que exploram comportamentos permitem explorar algum caso de uso, história do usuário ou cenário de aplicação complexo em um sistema.

Para executar a prova de conceito, foram planejados um estudo piloto com um *workflow* fictício, denominado "SimpleCount" e um segundo com *workflows* reais disponibilizados no repositório myExperiment. Essa prova de conceito foi utilizada para embasar a capacidade de gerenciar os experimentos científicos e os *workflows* relacionados, afim de analisar o uso dos serviços web, o modelo de armazenamento dos dados e as regras e restrições existentes na ontologia PROV-OEXT.

Apesar de não apresentar o formalismo de um estudo experimental, a realização de provas de conceitos contribui para a análise da arquitetura. Além disso, possibilitou também a verificação da viabilidade da solução proposta nesta dissertação. Na seção 5.3 é apresentada a análise do repositório, enfatizando seus objetivos e as dificuldades encontradas. Como base nos resultados da análise realizada em 5.3, a seção 5.4 detalha a prova de conceito, com o objetivo de analisar as funcionalidades da arquitetura.

5.3 AVALIAÇÃO DAS CARACTERÍSTICAS DE MANUTENÇÃO E EVOLUÇÃO EM REPOSITÓRIOS DE WORKFLOWS EXISTENTES

No contexto de manutenção e evolução de software, uma fonte importante de informação são os dados históricos e, como já mencionado, a falta desse tipo de informação dificulta a análise dos resultados do experimento e dos *workflows* que o compõem. Com o objetivo de mapear dados históricos, extraídos de repositórios existentes, foi realizado um estudo a fim de analisar a manutenção e evolução dos *workflows* armazenados. O referido estudo foi realizado junto aos repositórios myExperiment e CrowDLabs. No repositório CrowDLabs não foi possível extrair nenhum dado devido a necessidade de permissão de acesso a base, desse modo, para essa avaliação, foi utilizado então o repositório myExperiment.

Como primeiro passo na avaliação do myExperiment, foi selecionado um conjunto de *workflows* do repositório que atendiam aos seguintes critérios: i) serem modelados nos principais SGWfC, i.e., Taverna, Kepler e VisTrails; ii) terem sido disponibilizados por pesquisadores que contribuem de maneira preponderante disponibilizando *workflows* para o repositório; iii) serem relacionados a um maior número de *workflows* similares, de forma a enfatizar as características de manutenção. Foram selecionados *workflows* ao longo do mês de abril de 2015, considerando os critérios apresentados. Com isso, foram selecionados 254 *workflows*, sendo 4 do VisTrails, 47 do Kepler e 203 do Taverna 2 [5]. Com exceção do Taverna, cujo critério de seleção foi baseado prioritariamente no critério ii), ou seja, foram selecionados os *workflows* com base nos pesquisadores que possuem o maior número de *workflows* disponíveis na base myExperiment, os demais foram selecionados com base somente nos critérios i) e iii), uma vez que eram poucos *workflows*. Foram utilizados *workflows* do VisTrails, Kepler e Taverna nas versões 2.2.3, 2.5 e 2.5, respectivamente.

Considerando esse conjunto de *workflows*, foi proposto verificar o total de abordagens descendentes que a arquitetura E-SECO ProVersion identificava, onde buscou-se analisar se os pesquisadores que disponibilizaram *workflows* no repositório utilizavam a abordagem descendente para a construção de novos *workflows*. Se sim, o uso da arquitetura E-SECO ProVersion, conjugada com o repositório myExperiment, traria indícios de um melhor controle da gestão desta informação, uma vez que é possível o armazenamento da proveniência do *workflow* a partir do uso da E-SECO ProVersion, o que contribui para a validação do *workflow* e possibilita sua reutilização em outras pesquisas e também por outros pesquisadores, uma vez que o mesmo possui conhecimento sobre como o *workflow* foi criado e mantido ao longo do tempo. Caso não fosse possível, o uso da E-SECO ProVersion poderia auxiliar na descoberta de informações de manutenção e evolução entre os *workflows* armazenados.

Considerando essas premissas, foi realizada então uma análise dos *workflows* selecionados. Durante a execução dos *workflows* do VisTrails observou-se que nenhum dos 4 *workflows* disponíveis estava funcionando, pois todos utilizavam um mesmo serviço web que não estava mais ativo. Com isso, não foi possível capturar os dados dos *workflows* e realizar a verificação, visto que nenhum dos *workflows* disponíveis estava funcionando.

[5] *Workflows* do Taverna 1 apresentam alguns problemas de compatibilidade com a segunda versão do SGWfC do Taverna.

Esta falta de informação sobre o serviço indisponível dificulta o reuso dos *workflows* e a proposição de um serviço que possa ser o substituto do que estava "quebrado".

Nos *workflows* do SGWfC Kepler, foi verificado que, dos 47 *workflows* obtidos, apenas 1 possuía informação de versionamento, indicada pelo pesquisador, informando que houve uma correção na construção do mesmo. Com isso, restaram 46 *workflows* diferentes no repositório, o que representa apenas 2% do total de *workflows*, conforme apresenta a Figura 5.22. Isso demonstra que poucas informações são disponibilizadas sobre os *workflows* o que dificulta sua reutilização. Assim, como dito acima, isso caracteriza uma oportunidade de utilização da arquitetura E-SECO ProVersion, no sentido de capturar mais informações a respeito dos *workflows* e, através de inferências, a partir do uso da ontologia, derivar informações que permitam identificar características de versionamento, manutenção e evolução.

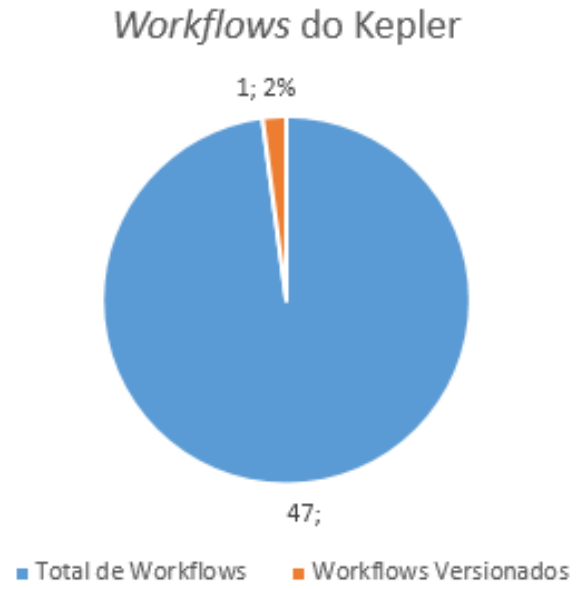


Figura 5.22: Análise dos *workflows* do Kepler.

Continuando a análise, dos 46 *workflows* diferentes obtidos, apenas 26 encontravam-se funcionando e, ao executar a E-SECO ProVersion, observou-se que, no total, 16 *workflows* apresentavam alguma característica de similaridade, o que representa 61,54% dos *workflows* disponíveis e funcionando na base, indicando características de abordagem descendente, manutenção ou evolução, conforme Figura 5.23. Assim, com o uso da arquitetura E-SECO ProVersion, conjugada ao repositório, o pesquisador passa a ter controle dos dados históricos do *workflow*, o que contribui para a criação de novos *workflows* utilizando abordagens descendentes ou mesmo na reutilização de um *workflow* existente.

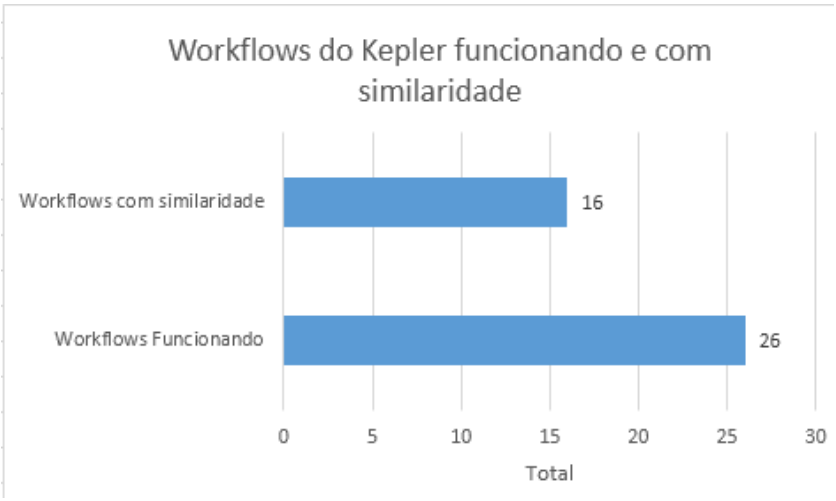


Figura 5.23: Análise dos *workflows* do Kepler com relação a funcionamento e similaridade.

Já na análise dos *workflows* do SGWfC Taverna, foram recuperados, conforme dito, 203 *workflows* e a primeira análise realizada foi para identificar quais estavam funcionando. Observou-se que apenas 73 *workflows* do total de 203 estavam funcionando, o que corresponde a 35,96% conforme Figura 5.24. Essa informação é útil para que o pesquisador saiba se o *workflow* pode ser reutilizado em outro estudo ou dependerá de algum tipo de manutenção. Com o uso da E-SECO ProVersion essa informação é disponibilizada para o pesquisador de forma clara.

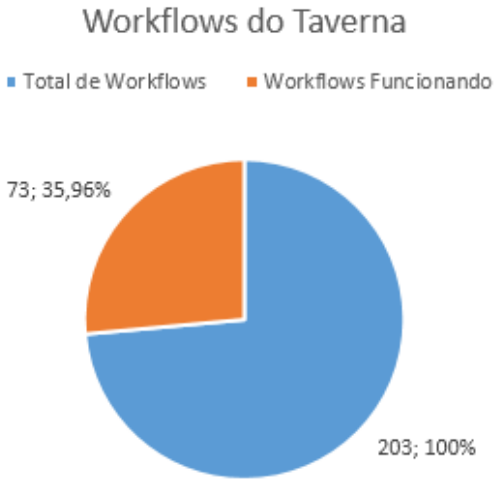


Figura 5.24: Análise dos *workflows* do Taverna.

Dos 73 *workflows*, 29 apresentavam versionamento indicado pelo pesquisador que o disponibilizou e a E-SECO ProVersion identificou ainda características de similaridade em 60 *workflows*, conforme Figura 5.25. Essa informação foi extraída da ontologia PROV-OEXT, com base nas tarefas em comum utilizadas pelos *workflows*, o que indica ainda

uma possibilidade de abordagem descendente na composição dos mesmos, facilitando o reuso destes *workflows* e a derivação de novos com base nessa linha de evolução. Um dos objetivos da E-SECO ProVersion é gerenciar os experimentos e os *workflows* vinculados, de modo que o pesquisador tenha pleno conhecimento dos mesmos.

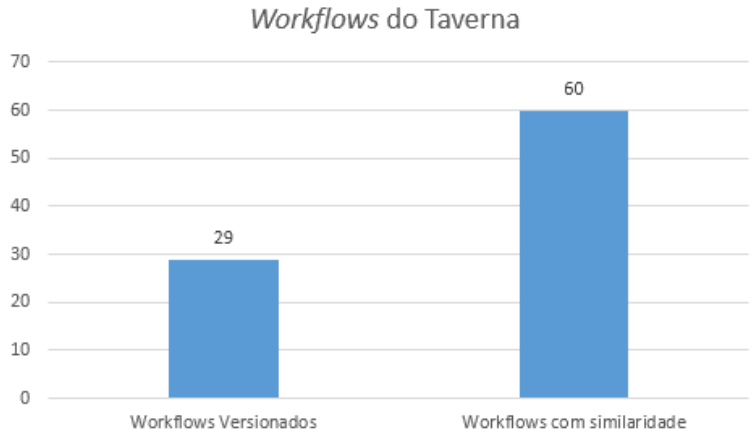


Figura 5.25: Análise dos *workflows* do Taverna com relação à similaridade.

Assim, com essa análise pode-se identificar que, dos 254 *workflows* avaliados, apenas 99 estavam funcionando, o que representa 38,98% do total, indicando que a maior parte apresentava algum problema, conforme Figura 5.26. Isso demonstra que para a reutilização desses *workflows*, o pesquisador necessitaria de realizar uma análise mais detalhada para manutenção dos mesmos. O que pode impactar no tempo total do experimento e em retrabalho pelo pesquisador. Com o uso da E-SECO ProVersion, tal informação é disponibilizada de maneira clara, facilitando a identificação da linha de evolução dos *workflows*, as manutenções sofridas e as oportunidades de reutilização dos mesmos.

Desses 99 *workflows* funcionando, apenas 30 possuíam histórico de versionamento, ou seja, 30,3% do total conforme Figura 5.27, o que indica um número baixo de informações disponibilizadas dos mesmos. Assim, com o uso da E-SECO ProVersion, conjugada com o myExperiment, o pesquisador tem acesso a informações que auxiliam no controle dos dados do experimento e das versões dos *workflows* vinculados aos experimentos, facilitando o trabalho do pesquisador e permitindo a formação de um banco de conhecimento sobre os experimentos executados e os *workflows* disponíveis, de forma a contribuir para a comparação entre resultados dos experimentos e reutilização dos *workflows*, como já mencionado.

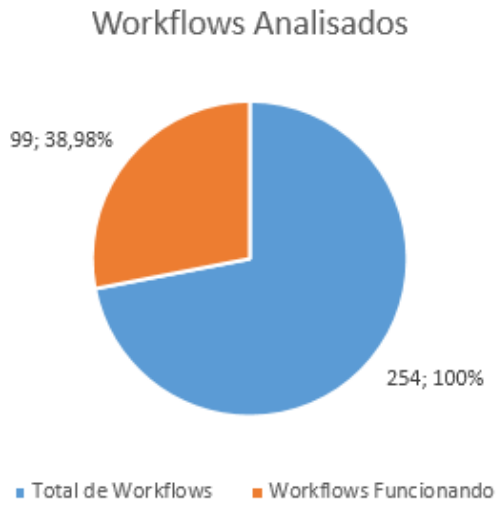


Figura 5.26: Análise total dos *workflows*.

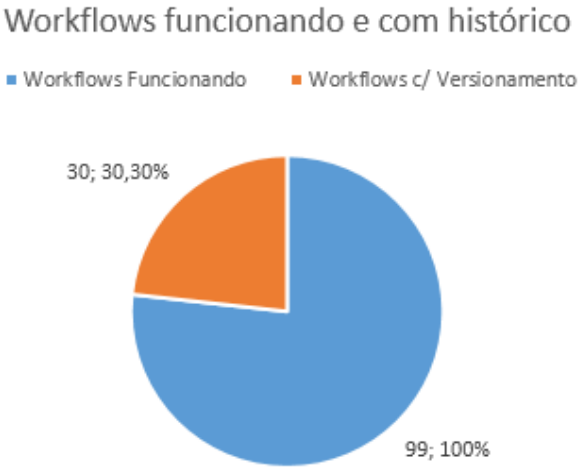


Figura 5.27: Análise total dos *workflows* funcionando e com versionamento.

De forma geral, com a utilização da arquitetura E-SECO ProVersion, foi possível identificar similaridade em 76,77% dos *workflows* funcionando, conforme Figura 5.28, demonstrando forte indício do uso da abordagem descendente na construção de novos *workflows*. Essa informação não está disponível atualmente no repositório myExperiment. Assim, considera-se que o uso da arquitetura E-SECO ProVersion mostra indícios de benefícios para o pesquisador, no que tange a identificação de características de manutenção e evolução.

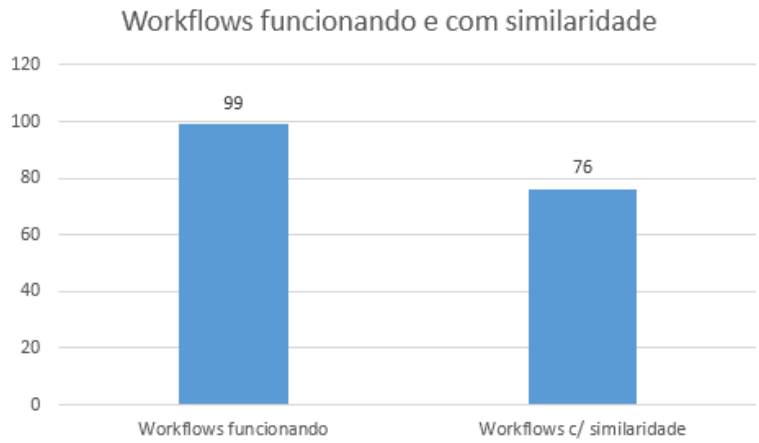


Figura 5.28: Análise total dos *workflows* funcionando e com similaridade.

Por fim, foram verificados os problemas mais comuns encontrados durante a execução dos *workflows*, a saber falhas de execução, serviços indisponíveis e falta de parâmetros de execução, *workflows* que encontravam-se duplicados na base do myExperiment, entre outros. O resultado dessas análises podem ser vistos nas figuras 5.29 e 5.30. A Figura 5.29 apresenta os 3 problemas mais comuns identificados: falhas de execução, serviços indisponíveis e falta de parâmetros de execução. Já a Figura 5.30 apresenta os *workflows* repetidos na base do myExperiment, totalizando 25 *workflows*, sendo identificados dentre os *workflows* analisados, 13 (52%) *workflows* duplicados, 5 (20%) *workflows* triplicados, 6 (24%) *workflows* quadruplicados e 1 (4%) *workflow* repetido na base 5 vezes. Isso demonstra uma falta de controle por parte dos pesquisadores, pois disponibilizam o mesmo *workflow* na base várias vezes, reforçando com isso a necessidade do uso da E-SECO ProVersion, pois permite a gerência dessas informações de forma otimizada.

Em resumo, verificando os resultados da análise realizada no repositório myExperiment, pode-se destacar que informações dos *workflows* que estão ativos são importantes para que o pesquisador tenha conhecimento sobre os *workflows* que estão disponíveis para utilização e quais dependem de algum tipo de manutenção, uma vez que grande parte dos *workflows* disponíveis no repositório do myExperiment não estão funcionando. Já a informação extraída de *workflows* repetidos no repositório reforça a necessidade do uso da E-SECO ProVersion, pois mostra que os pesquisadores desconhecem os *workflows* disponíveis e, com isso, ao invés de reutilizarem um *workflow* existente, acabam especificando um novo bastante similar. Já com o uso da E-SECO ProVersion, o pesquisador tem a possibilidade de identificar a existência de um *workflow* similar ao desejado, evitando

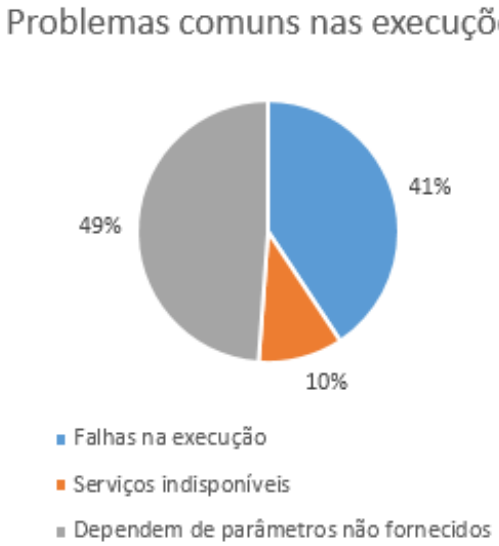


Figura 5.29: Problemas mais comuns na execução dos *workflows*.

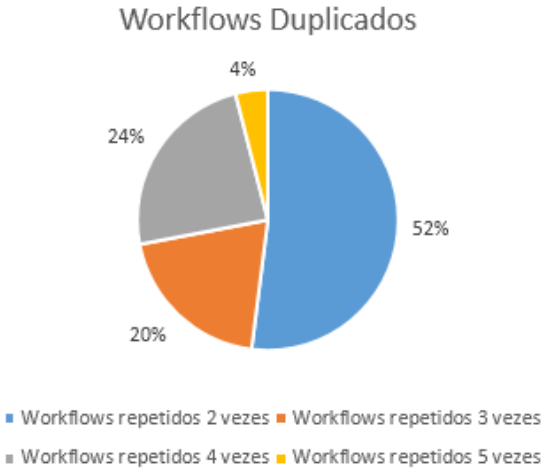


Figura 5.30: *Workflows* duplicados na base.

retrabalho na construção de um novo.

A partir dessa análise, pode-se destacar também que controlar os dados de execução de um experimento científico não é uma tarefa trivial e com o uso mecanismos como os da arquitetura E-SECO ProVersion, essa tarefa torna-se mais fácil, visto que a arquitetura permite gerenciar o experimento e os *workflows* existentes.

De maneira geral, essa avaliação permitiu identificar alguns problemas existentes no repositório myExperiment, que são: i) falta de informações históricas dos *workflows*; ii) falta de controle na composição de novos *workflows*; iii) falta de informações dos *workflows* possíveis de reutilização; iv) falta de informações de experimentos e seus *workflows* associados; e v) duplicação de *workflows* no repositório por falta de controle e conhecimentos

dos *workflows* existentes.

Apesar dessa avaliação apresentar resultados que demonstram a necessidade de controle dos experimentos e *workflows*, é adequado realizar uma prova de conceito de forma a apresentar em detalhes as funcionalidades da E-SECO ProVersion considerando *workflows* sem indicação de versionamento do myExperiment, para inferir características de manutenção e evolução.

5.4 PROVA DE CONCEITO

Assim, considerando os problemas encontrados na análise realizada na seção 5.3, optou-se por realizar uma prova de conceito (*Proof of Concept - PoC*) (CALDIERA; ROMBACH, 1994) , com o objetivo de detalhar o uso da arquitetura, enfatizando a captura dos dados e extração de informação com uso da ontologia. Foi utilizado um *workflow* criado pelo autor da arquitetura E-SECO ProVersion e três *workflows* disponíveis no repositório do myExperiment. O primeiro *workflow* foi modelado de forma a facilitar o uso de funcionalidades específicas da E-SECO ProVersion, i.e., uso de ontologia e mecanismos de inferência, tendo como finalidade a busca por *workflows* similares no repositório myExperiment. Os outros *workflows* foram selecionados do repositório myExperiment por apresentarem alguma semelhança com o anterior e características de evolução e de manutenção, mas não explicitamente declaradas nos seus metadados.

Para realizar a prova de conceito foi utilizada a abordagem *GQM* (*Goal/Question/-Metric*) (BASILI, 1993) e seu objetivo pode ser definido da seguinte forma:

"**Analisar** a arquitetura E-SECO ProVersion **com a finalidade de** avaliar a manutenção e evolução de experimentos científicos e *workflows* vinculados **sob o ponto de vista de** pesquisadores **no contexto de** ecossistemas de software científico".

Na seção 5.4.1 serão apresentados os detalhes do *workflow* utilizado como piloto na prova de conceito e as informações extraídas pela arquitetura. Já na seção 5.4.2 serão apresentados os *workflows* do myExperiment submetidos a E-SECO ProVersion e serão avaliados os benefícios obtidos pela arquitetura.

5.4.1 *WORKFLOW* SIMPLECOUNT

O *workflow* "SimpleCount"é um *workflow* simples, criado pelo desenvolvedor da arquitetura E-Seco ProVersion. O *workflow* é composto por 3 tarefas com operações matemáticas, sendo elas adição, subtração e multiplicação, e foi desenvolvido no SGWfC Kepler.

O objetivo é detalhar as funcionalidades da E-SECO ProVersion, com especial ênfase na captura dos dados de execução do experimento e a derivação de informações para o pesquisador, através do uso de ontologias e mecanismos de inferência. Além disso, tem-se como objetivo secundário, realizar a busca em repositórios de *workflows* (myExperiment, nesse caso), com o objetivo de identificar *workflows* semelhantes, considerando características de evolução ou manutenção.

Para a captura dos dados, o *workflow* foi instrumentado com o serviço web disponibilizado pela E-SECO ProVersion para coleta dos dados de proveniência, conforme pode ser visto na Figura 5.31. O serviço web é acoplado ao *workflow* de forma manual. Assim, pode-se vincular as tarefas que compõem o *workflow* ao serviço web, de forma que o serviço web passe a fazer a captura dos dados e esses sejam armazenados no banco de dados e, posteriormente, utilizados na ontologia PROV-OEXT.

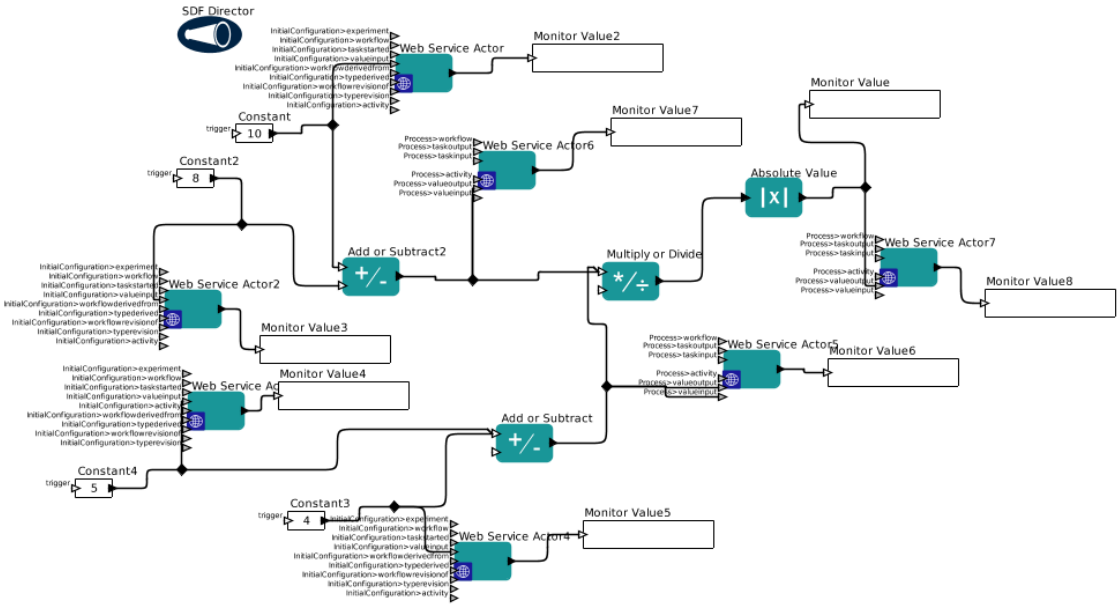


Figura 5.31: *Workflow* piloto.

Após a execução do *workflow* instrumentado, pelo SGWfC Kepler, os dados da execução ficaram disponíveis no repositório da plataforma E-SECO e o pesquisador pôde

consulta-los. Consultas simples como, por exemplo, tempo de execução de uma tarefa, não dependem do uso da ontologia e podem ser realizadas diretamente com os dados de execução armazenados, conforme Figura 5.32. Informações com maior nível de detalhamento como, por exemplo, informações de manutenção e evolução do *workflow*, experimentos ou atividades em que o mesmo possui influência, necessitam do uso da ontologia PROV-OEXT. Para isso, é necessário que o pesquisador selecione a opção "Load Ontology"e os dados do repositório sejam carregados para a ontologia e geradas as inferências.

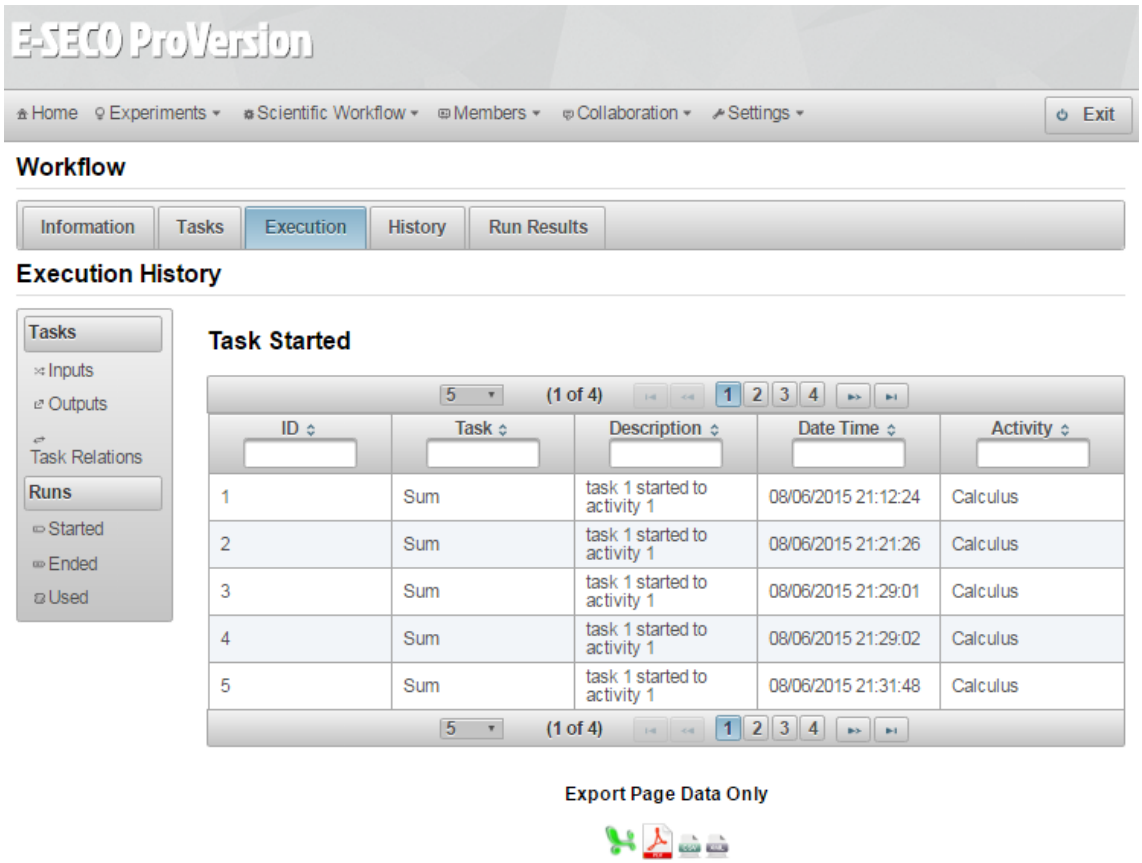


Figura 5.32: Consulta das informações do *workflow* piloto na base da E-SECO.

A Figura 5.33 apresenta as informações inferidas na ontologia sobre o *workflow* "SimpleCount", onde pode-se visualizar que o mesmo foi executado no SGWfC Kepler, logo sofreu influência do mesmo. Também é possível visualizar que o *workflow* usou 3 tarefas do experimento "Mathematical.operations"e, com isso, foi influenciado por elas. Por fim, a ontologia identificou que existe um *workflow* similar ao "SimpleCount"denominado "SimpleCount2"e que está na versão 01.00.00, sendo que o *workflow* "SimpleCount2"também é um *workflow* de operações matemáticas e havia sido construído para teste da arquitetura.

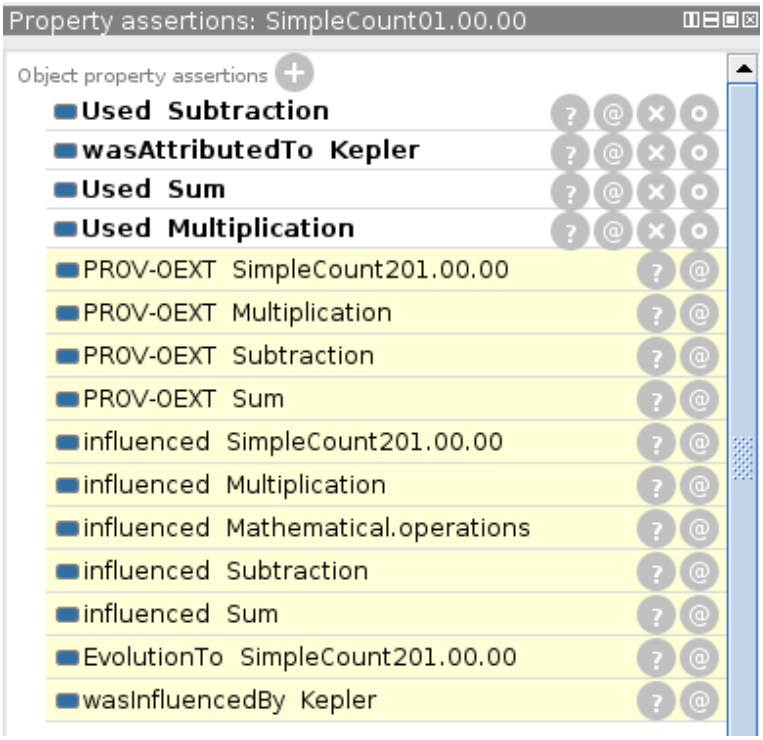


Figura 5.33: Consulta das informações do *workflow* piloto na ontologia.

Além da derivação de conhecimento novo, foi realizado o acesso ao repositório do myExperiment para busca por *workflows* similares. Isso é importante para que o pesquisador possa conhecer alternativas ao *workflow* em uso ou mesmo evitar retrabalho na composição do mesmo. Assim, ao invés do pesquisador ter o trabalho de fazer a alteração em um serviço que deixou de funcionar, pode verificar se algum outro pesquisador que faça uso do mesmo *workflow* já o tenha feito, ou mesmo após a modelagem abstrata do *workflow*, na etapa de prototipação do experimento.

Além disso, a arquitetura permite ao pesquisador verificar se algum outro pesquisador já tenha disponibilizado um *workflow* semelhante ao desejado. Isso é realizado através da comparação das tarefas utilizadas no *workflow*. A arquitetura apresenta todos os *workflows* similares, conforme Figura 5.34. Essa busca por *workflows* similares é feita através de uma string de busca executada sob uma API do repositório myExperiment existente na plataforma E-SECO, onde são passados como parâmetros o número de tarefas e quais são. Os *workflows* retornados possuem o mesmo número de tarefas do selecionado, conforme pode ser observado na Figura 5.34.

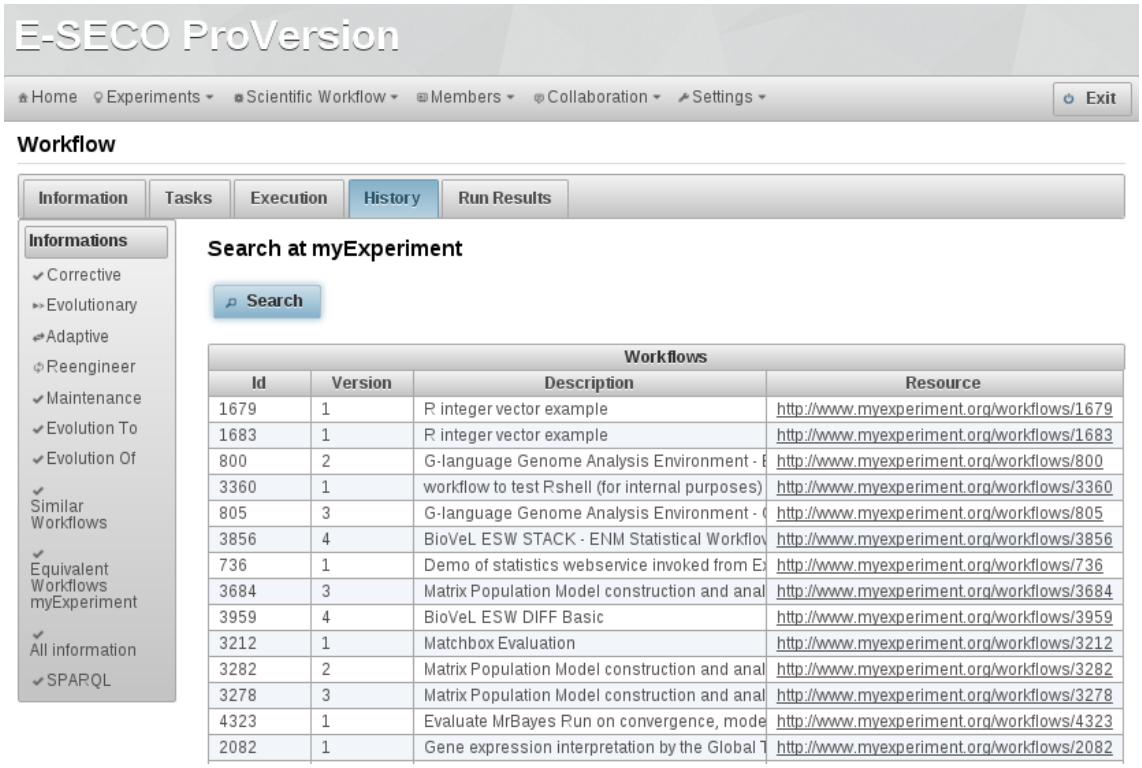


Figura 5.34: Busca por *workflows* similares no myExperiment.

Como resultado da busca no myExperiment foram exibidas as informações de ID do *workflow* no repositório, quantidade de versões do mesmo *workflow* disponível, uma descrição e o endereço de recuperação do *workflow*, conforme Figura 5.34.

O *workflow* "SimpleCount"foi desenvolvido para uma avaliação piloto e o mesmo não segue uma abordagem descendente, ou seja, não foi criado com base em um *workflow* real, sendo utilizado como base para encontrar *workflows* semelhantes que pudessem demonstrar as características desejadas deste trabalho.

Nessa prova de conceito foi possível apresentar as informações que são capturadas, as que são inferidas pela ontologia PROV-OEXT e com base na seleção de um *workflow*, como a arquitetura permite identificar outros *workflow* semelhantes no repositório. Entretanto, não foi possível demonstrar a identificação de versionamento entre *workflows* pela arquitetura e a identificação dos *workflows* com necessidade de manutenção, o que será feito na prova de conceito da seção 5.4.2, com *workflows* reais do myExperiment.

Assim, para complementar a PoC, foram selecionados 3 *workflows* do repositório myExperiment que serão apresentados na seção seguinte. A escolha dos dois primeiros *workflows* se deve ao fato de possuírem características de evolução, indicando uma

abordagem descendente na construção dos mesmos, extraída na avaliação apresentada na seção 5.3 e não possuírem informações de versionamento entre si. Já o último *workflow* foi escolhido por apresentar um serviço que não estava mais disponível, indicando a necessidade de manutenção, também observado na avaliação da seção 5.3.

5.4.2 *WORKFLOW*S DO MYEXPERIMENT

Para avaliar a abordagem da arquitetura E-SECO ProVersion em identificar abordagens descendentes e versionamento entre os *workflows*, foram selecionados 2 *workflows* no myExperiment, sendo o primeiro denominado "Álgebra 1 - IST 600"[6] e o segundo "Álgebra 2 - IST 600"[7]. Ambos os *workflows* foram desenvolvidos pelo mesmo pesquisador identificado como "Gstalloch"[8] e não possuíam indicação de versionamento ou derivação.

O primeiro *workflow* (Álgebra 1 - IST 600) apresenta um conjunto com 3 tarefas, todas relacionadas a operações matemáticas e foi modelado no SGWfC Kepler, conforme Figura 5.35.

O segundo *workflow* (Álgebra 2 - IST 600) também desenvolvido para o SGWfC Kepler, realiza operações matemáticas e possui um conjunto de 7 tarefas, conforme Figura 5.36.

Ao instanciar cada um dos *workflows*, os dados da execução foram coletados pela E-SECO ProVersion, utilizando o serviço web disponível. Como primeiro resultado, a arquitetura E-SECO ProVersion foi capaz de identificar características de semelhança entre ambos. Observou-se ainda que ambos utilizavam do mesmo conjunto de tarefas, os parâmetros de entrada eram similares em ambos e o *workflow* "Álgebra 1 - IST 600" era parte do *workflow* "Álgebra 2 - IST 600".

Essas características de ambos os *workflows* foram inferidas com o uso da ontologia PROV-OEXT e apresentadas pelo módulo de manutenção e evolução, sendo que, com base no grau de semelhança, a E-SECO ProVersion pode indicar qual *workflow* evoluiu do outro, caracterizando o uso de uma abordagem descendente. Nesse caso, como o *workflow* "Álgebra 1 - IST 600" é parte do *workflow* "Álgebra 2 - IST 600", indica-se que

[6]http://www.myexperiment.org/workflows/2430.html
[7]http://www.myexperiment.org/workflows/2431.html
[8]http://www.myexperiment.org/users/18929.html

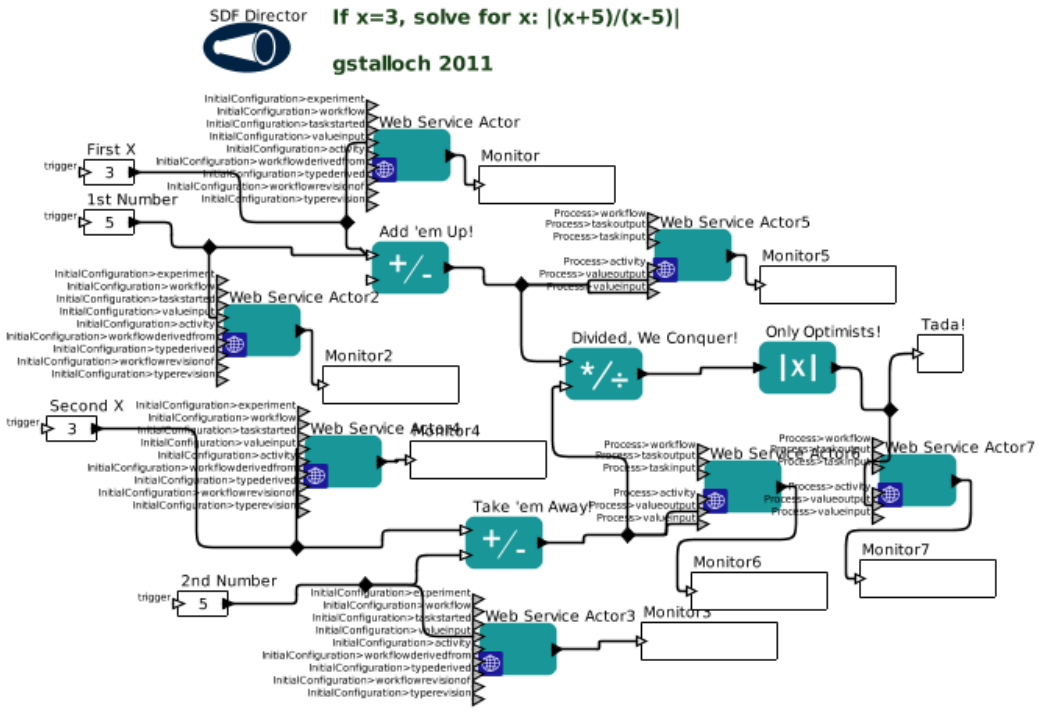


Figura 5.35: *Workflow* Álgebra 1 - IST 600.

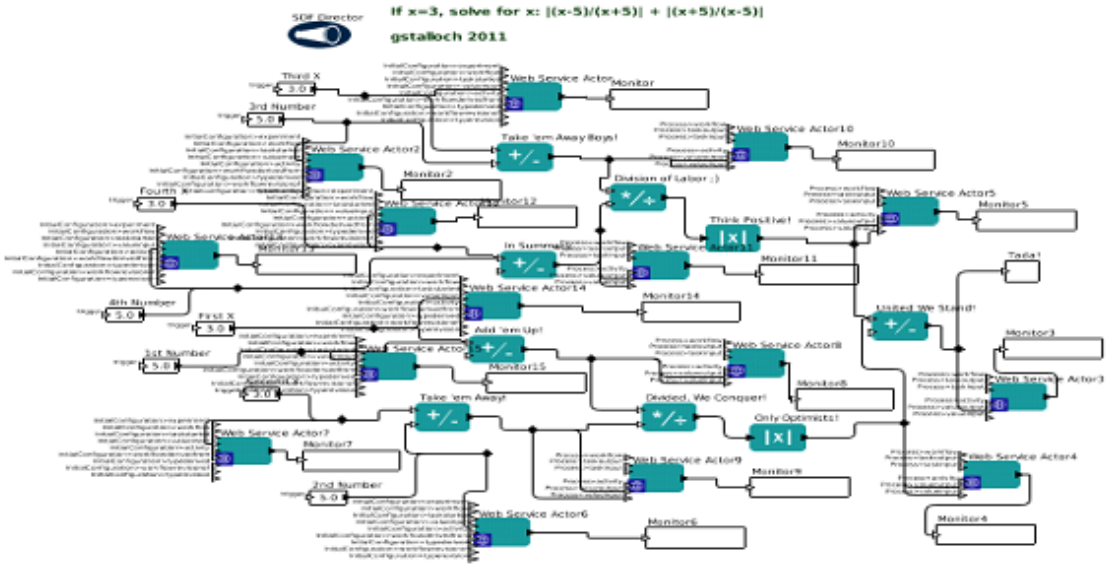


Figura 5.36: *Workflow* Álgebra 2 - IST 600.

o *workflow* "Álgebra 2 - IST 600" foi desenvolvido com base no "Álgebra 1 - IST 600", caracterizando uma abordagem evolutiva entre o "Álgebra 1 - IST 600"e o "Álgebra 2 - IST 600", conforme Figura 5.37.

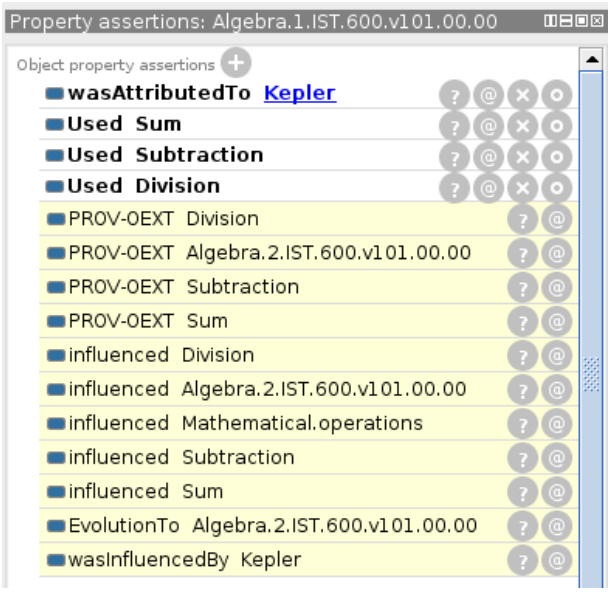


Figura 5.37: Inferências do *workflow* Álgebra 1 - IST 600.

A informação de semelhança entre os *workflows* é apresentada pela arquitetura E-SECO ProVersion de forma clara ao pesquisador, dispensando que o mesmo tenha conhecimento sobre ontologias, conforme Figura 5.38. Com isso, pode-se observar uma linha evolutiva do *workflow* "Algebra 1 - IST 600" para o "Algebra 2 - IST 600". Logo, ao modificar uma tarefa constante no primeiro *workflow*, a arquitetura deve indicar a necessidade de manutenção em ambos os *workflows*.

Outra característica da E-SECO ProVersion é, ao instanciar o *workflow* e, por meio dos dados coletados, identificar se ocorreu alguma falha e permitir ao pesquisador verificar onde ocorreu. Para apresentar essa funcionalidade, foi utilizado um dos *workflows* selecionados na seção 5.3, também disponível no repositório myExperiment. Esse *workflow* pode ser visto na Figura 5.39 e é denominado "Extract Gene Sequence with Kepler"[9] onde o mesmo, ao tentar ser instanciado, apresenta erro no acesso ao serviço REST utilizado e a arquitetura E-SECO ProVersion registra isso, indicando que o serviço deve ser substituído por outro similar. Essa informação de falha em tarefas do *workflow* são exibidas na tela de resultados da E-SECO ProVersion, listando todas as tarefas e o resultado de cada uma (sucesso ou falha), conforme foi apresentado no roteiro de uso do E-SECO (seção 5.1), Figura 5.1.

[9]http://www.myexperiment.org/workflows/2459.html

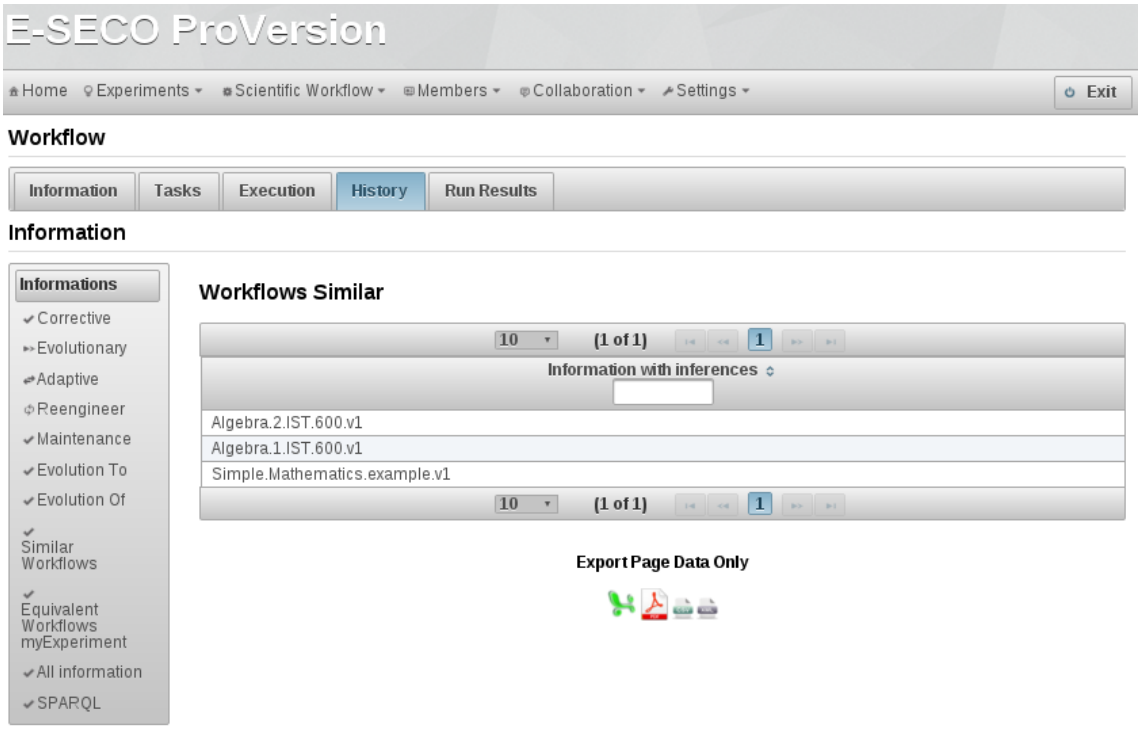


Figura 5.38: Similaridade dos *workflows* "Álgebra 1 - IST 600" e "Álgebra 2 - IST 600".

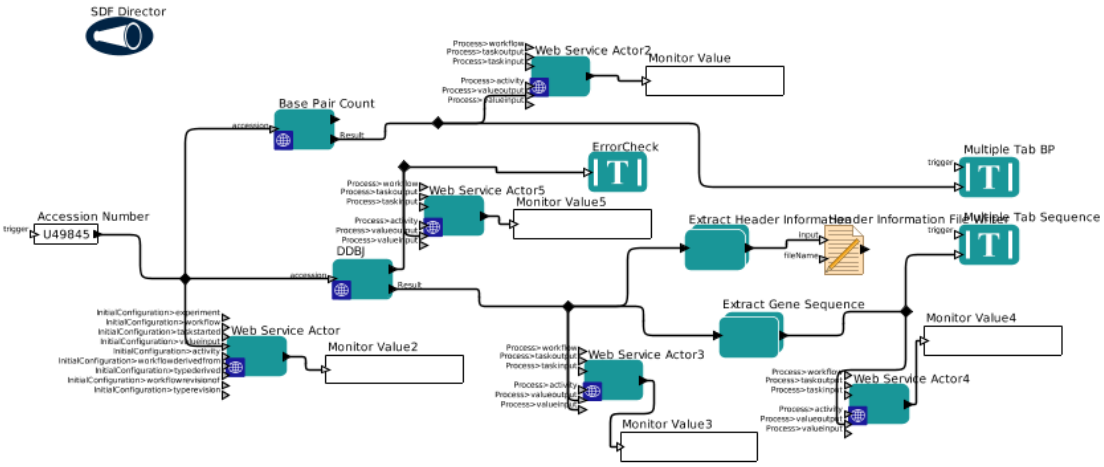


Figura 5.39: *Workflows* "Extract Gene Sequence with Kepler".

Com isso, o pesquisador pode verificar o que ocorreu durante a execução de cada tarefa. Além disso, a E-SECO ProVersion registra os dados de entrada, saída e o tempo de execução das tarefas. Desta forma, o pesquisador pode verificar se os dados estão corretos e a ocorrência de alguma anomalia na execução que causou algum descontrole na execução do experimento.

Caso o pesquisador precise substituir o serviço utilizado no *workflow*, como no exemplo

do “Extract Gene Sequence with Kepler”, a E-SECO ProVersion permite ao pesquisador realizar a busca na base do BioCatalogue, por exemplo, ou então no repositório de serviços da E-SECO, conforme Figura 5.40, facilitando o trabalho do pesquisador na seleção de serviços para a composição do *workflow*, como já demonstrado.

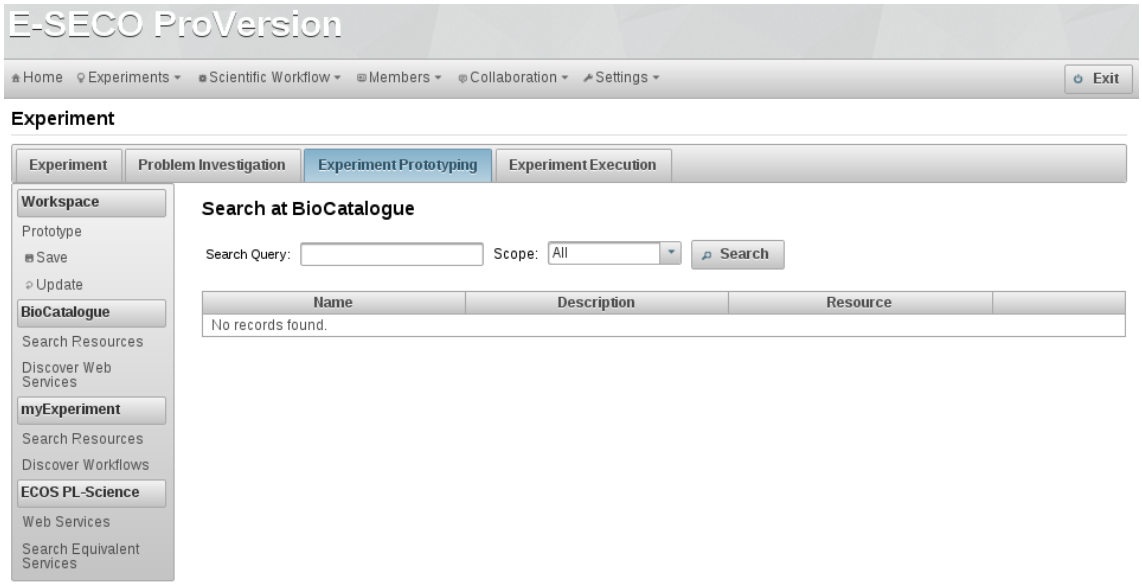


Figura 5.40: Acesso ao BioCatalogue.

5.5 DISCUSSÕES

Conforme apresentado nas seções 5.3 e 5.4, a arquitetura E-SECO ProVersion auxilia o pesquisador na gerência de configuração dos experimentos científicos e dos *workflows* envolvidos, por meio da captura de dados de proveniência e uso de ontologia, tratando as manutenções e evoluções desses experimentos e *workflows* ao longo dos ciclos de experimentação.

Com as análises realizadas em 5.3, pode-se verificar a falta de controle na criação de novos *workflows* e a ausência de informações históricas no repositório myExperiment. Informações históricas são importantes para a pesquisa, pois permitem rastrear os passos que foram executados e criar uma base de conhecimento para posterior análise desses passos. Essa base de conhecimento é importante, uma vez que o pesquisador precisa validar a pesquisa e, nesse contexto, informações sobre a origem da pesquisa, seu andamento e mudanças ocorridas são preponderantes para o planejamento dos próximos passos e análises dos resultados produzidos.

Na seção 5.4 foi detalhada uma PoC de forma a verificar o uso da E-SECO ProVersion na derivação de conhecimento implícito, quando dados históricos não estão disponíveis. Nesse contexto, foi utilizada inferência conjuntamente com uma ontologia para derivar conhecimento sobre manutenções e características de evolução entre *workflows*. Nesse sentido, pode-se verificar que a arquitetura E-SECO ProVersion mostra indícios de que possa ser utilizada por pesquisadores para derivação de informações de manutenção e evolução. No entanto, uma avaliação rigorosa nesse sentido deve ser realizada, de forma a validar essa questão.

5.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou algumas diretrizes para uso da arquitetura E-SECO ProVersion, considerando *workflows* extraídos do repositório myExperiment e outros especificados pelos autores da proposta, com o intuito de apresentar de maneira detalhada as funcionalidades da arquitetura.

Com esse objetivo, foi detalhado um roteiro de uso da E-SECO ProVersion, utilizando um conjunto de 5 *workflows*, com o intuito de apresentar as funcionalidades da E-SECO ProVersion e como a arquitetura auxilia o pesquisador no gerenciamento de experimentos e *workflows* vinculados ao mesmo.

Além disso, também foi especificado um estudo que analisou o uso da arquitetura com repositórios de *workflows* científicos, com o objetivo de verificar a capacidade da arquitetura em identificar *workflows* com características similares. Para esse estudo foi utilizado o repositório myExperiment utilizando *workflows* reais. Os resultados preliminares se mostraram promissores em direção aos benefícios que a arquitetura, conjugada com repositórios de *workflows* existentes, pode trazer para a gerência de configuração de experimentos científicos. Não entanto, uma avaliação formal deve ser realizada para a comprovação dos resultados.

Por fim, uma PoC foi desenvolvida com o objetivo de apresentar o uso de ontologias conjugadas com máquinas de inferência para descoberta de conhecimento sobre manutenção e evolução em *workflows* existentes. Os resultados se mostram promissores mas uma avaliação formal deve ser conduzida.

No capítulo seguinte são apresentadas as conclusões e as indicações de trabalhos futuros que podem ser desenvolvidos.

6 CONSIDERAÇÕES FINAIS

Este trabalho detalhou a arquitetura E-SECO ProVersion, no contexto da plataforma de ecossistema E-SECO. A E-SECO ProVersion tem como objetivo identificar informações sobre manutenção e evolução de experimentos e *workflows* científicos e, com base nestas informações, auxiliar os pesquisadores nas análises dos dados dos experimentos. Acreditamos que o resultado destas análises pode auxiliar no reuso, manutenção e evolução, tanto dos experimentos como dos *workflows* científicos, de forma a reduzir o trabalho e proporcionar um maior controle da pesquisa.

A partir da revisão sistemática apresentada no capítulo 3, alguns resultados e desafios foram identificados na área, reforçando a necessidade de desenvolvimento de uma arquitetura de suporte a manutenção e evolução de *workflows* científicos. Assim, no capítulo 4 foi apresentada a arquitetura E-SECO ProVersion para apoio a manutenção e evolução de experimentos e *workflows* científicos, identificando as principais funcionalidades e desafios. Com o desenvolvimento da arquitetura, algumas diretrizes de uso foram apresentadas no capítulo 5, gerando evidências preliminares relacionadas aos ganhos com o uso da arquitetura, conforme destacado no capítulo.

Neste contexto, algumas contribuições da arquitetura E-SECO ProVersion podem ser destacadas:

i Captura dos dados de execução do *workflow*, através de um serviço web, permitindo que a arquitetura seja independente do SGWfC utilizado pelo pesquisador;

ii Uso do modelo de proveniência PROV modelado para atender aos requisitos de gerenciamento de experimentos e *workflows* científicos, permitindo ao pesquisador ou grupo de pesquisa criar uma base histórica com dados de proveniência sobre os experimentos científicos e *workflows* utilizados;

iii Desenvolvimento da ontologia PROV-OEXT para extração de conhecimento implícito, considerando a base de dados de proveniência desenvolvida em (ii);

iv Interface para análise dos dados de proveniência sem a necessidade de conhecimento

sobre banco de dados ou ontologias, o que é condição importante para cientistas de diferente domínios que não o da ciência da computação;

v Integração com serviços BioCatalogue e myExperiment, auxiliando o pesquisador na composição de novo *workflow* ou na reutilização de um existente;

6.1 LIMITAÇÕES

Ao decorrer deste trabalho foram encontrados algumas limitações, principalmente relacionadas a pouca disponibilidade de dados históricos e a reuso de *workflows*. A seguir destacamos as mais importantes:

- Há falta de dados históricos relacionados a *workflows* científicos de maneira geral, conforme apresentado na seção 4.5, o que dificultou a análise da arquitetura E-SECO ProVersion em identificar e classificar a manutenção e evolução dos mesmos;
- Uma avaliação mais criteriosa da arquitetura também não pôde ser realizada, justamente pela falta de dados históricos. Somente algumas diretrizes de uso foram detalhadas, utilizando *workflows* existentes no repositório myExperiment e alguns criados pelo autor da pesquisa. Assim, um experimento formal, com critérios bem definidos deve ser especificado, de forma a avaliar a aplicabilidade da arquitetura E-SECO ProVersion em cenários reais.
- Como a captura dos dados de execução e realizada através do serviço web disponível na arquitetura, é necessário que o pesquisador acople o serviço ao workflow antes de sua execução pelo SGWfC. Apesar desta limitação para uso da arquitetura, possibilita que o E-SECO ProVersion possa ser utilizado por pesquisador de forma independente do SGWfC em uso.

Apesar destas limitações, o presente trabalho buscou explorar de forma detalhada as características de gerenciamento de configuração dos experimentos e *workflows* científicos através do uso de dados de manutenção e evolução.

6.2 TRABALHOS FUTUROS

A arquitetura E-SECO ProVersion é uma etapa inicial para a gerência de configuração de experimentos e *workflows* científicos no contexto da plataforma E-SECO. Desta forma, novas características de gerencia de configuração podem ser incluídas, tratando dos experimentos, *workflows* e considerando o contexto de colaboração, distribuição e heterogeneidade da plataforma E-SECO.

Algumas das características que podem ser melhor exploradas estão:

- Formação de um repositório de dados históricos e a partir deste, evoluir as características de manutenção e evolução hoje consideradas na abordagem. Esse é um ponto primordial para melhoria da abordagem, considerando inclusive a possibilidade de condução de um experimento formal para sua avaliação;
- Extensão das regras ontológicas, de forma a englobar novas características ainda não observadas, como a colaboração entre os grupos de pesquisa;
- Melhoria nas formas de visualização dos dados de manutenção e evolução, considerando tanto os dados de proveniência, quanto os dados das ontologias e as possíveis inferências.

REFERÊNCIAS

ALTINTAS, I. *et al.* Kepler: an extensible system for design and execution of scientific workflows. In: IEEE. **Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on**. [S.l.], 2004. p. 423–424.

ALTINTAS, I. *et al.* Camera 2.0: A data-centric metagenomics community infrastructure driven by scientific workflows. In: IEEE. **Services (SERVICES-1), 2010 6th World Congress on**. [S.l.], 2010. p. 352–359.

ANTONIOU, G.; HARMELEN, F. V. **A semantic web primer**. [S.l.]: MIT press, 2004.

BARGA, R. *et al.* The trident scientific workflow workbench. In: IEEE. **eScience, 2008. eScience'08. IEEE Fourth International Conference on**. [S.l.], 2008. p. 317–318.

BARGA, R. S.; DIGIAMPIETRI, L. A. Automatic capture and efficient storage of e-science experiment provenance. **Concurrency and Computation: Practice and Experience**, Wiley Online Library, v. 20, n. 5, p. 419–429, 2008.

BASILI, V. R. Applying the goal/question/metric paradigm in the experience factory. **Software Quality Assurance and Measurement: A Worldwide Perspective**, London, UK: Chapman and Hall, p. 21–44, 1993.

BASTOS, B. F.; BRAGA, R. M. M.; GOMES, A. T. A. Scientific workflow interchanging through patterns: Reversals and lessons learned. In: IEEE. **e-Science (e-Science), 2015 IEEE 11th International Conference on**. [S.l.], 2015. p. 557–564.

BELHAJJAME, K. *et al.* User feedback as a first class citizen in information integration systems. In: **CIDR**. [S.l.: s.n.], 2011. p. 175–183.

BELL, J. *et al.* Software design for reliability and reuse: a proof-of-concept demonstration. In: ACM. **Proceedings of the conference on TRI-Ada'94**. [S.l.], 1994. p. 396–404.

BELLOUM, A. *et al.* Collaborative e-science experiments and scientific workflows. **Internet Computing, IEEE**, IEEE, v. 15, n. 4, p. 39–47, 2011.

BHAGAT, J. *et al.* Biocatalogue: a universal catalogue of web services for the life sciences. **Nucleic acids research**, Oxford Univ Press, p. 394, 2010.

BIGARET, S.; MEYER, P. Bioside: from bioinformatics needs to a generic workflow engine. In: **2nd Decision Deck Developers Days**. [S.l.: s.n.], 2008.

BRERETON, P. *et al.* Lessons from applying the systematic literature review process within the software engineering domain. **Journal of systems and software**, Elsevier, v. 80, n. 4, p. 571–583, 2007.

CALDIERA, V.; ROMBACH, H. D. The goal question metric approach. **Encyclopedia of software engineering**, v. 2, n. 1994, p. 528–532, 1994.

CALLAHAN, S. P. *et al.* Managing the evolution of dataflows with vistrails. In: IEEE. **Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on**. [S.l.], 2006. p. 71–71.

CHAIKALIS, T. *et al.* Seagle: Effortless software evolution analysis. In: IEEE. **2014 IEEE International Conference on Software Maintenance and Evolution (ICSME)**. [S.l.], 2014. p. 581–584.

CIA, T. M. **Modelo de avaliação do processo de gerência de configuração de software**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação - USP, 2006.

COSTA, G. C. B. *et al.* A scientific software product line for the bioinformatics domain. **Journal of biomedical informatics**, Elsevier, v. 56, p. 239–264, 2015.

CRUZ, T. Uso e desuso de sistemas de workflow: Porque as organizações não conseguem obter retorno, nem sucesso, com investimentos em projetos de workflow. **Rio de Janeiro: e-papers**, 2006.

CUEVAS-VICENTTIN, V. *et al.* Modeling and querying scientific workflow provenance in the d-opm. In: IEEE. **High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:**. [S.l.], 2012. p. 119–128.

CUEVAS-VICENTTÍN, V. *et al.* The pbase scientific workflow provenance repository. **International Journal of Digital Curation**, v. 9, n. 2, p. 28–38, 2014.

DANTAS, C. Gerência de configuração de software. **Engenharia de Software Magazine**, v. 1, n. 2, 2009.

DEELMAN, E. *et al.* Pegasus: Mapping scientific workflows onto the grid. In: SPRINGER. **Grid Computing**. [S.l.], 2004. p. 11–20.

DEELMAN, E. *et al.* Workflows and e-science: An overview of workflow system features and capabilities. **Future Generation Computer Systems**, Elsevier, v. 25, n. 5, p. 528–540, 2009.

DEELMAN, E.; GIL, Y. Managing large-scale scientific workflows in distributed environments: Experiences and challenges. In: **e-Science**. [S.l.: s.n.], 2006. p. 144.

DIAMANTINI, C.; POTENA, D.; STORTI, E. Mining usage patterns from a repository of scientific workflows. In: ACM. **Proceedings of the 27th Annual ACM Symposium on Applied Computing**. [S.l.], 2012. p. 152–157.

DIAS, J. F. **Execução Interativa de Experimentos Científicos Computacionais em Larga Escala**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2013.

ESTUBLIER, J. Software configuration management: a roadmap. In: ACM. **Proceedings of the Conference on the Future of Software Engineering**. [S.l.], 2000. p. 279–289.

FREIRE, J. *et al.* Managing rapidly-evolving scientific workflows. In: **Provenance and Annotation of Data**. [S.l.]: Springer, 2006. p. 10–18.

FREITAS, V. *et al.* Uma arquitetura para ecossistema de software científico. **WDES 2015**, p. 41, 2015.

GASPAR, W. *et al.* Scientific provenance metadata capture and management using semantic web. **International Journal of Metadata, Semantics and Ontologies**, Inderscience Publishers (IEL), v. 10, n. 2, p. 123–138, 2015.

GIL, Y. *et al.* Wings for pegasus: Creating large-scale scientific applications using semantic representations of computational workflows. In: **Proceedings of the National Conference on Artificial Intelligence**. [S.l.: s.n.], 2007. v. 22, n. 2, p. 1767.

GOBLE, C. A. *et al.* myexperiment: a repository and social network for the sharing of bioinformatics workflows. **Nucleic acids research**, Oxford Univ Press, v. 38, n. suppl 2, p. W677–W682, 2010.

GOBLE, C. A.; ROURE, D. C. D. myexperiment: social networking for workflow-using e-scientists. In: ACM. **Proceedings of the 2nd workshop on Workflows in support of large-scale science**. [S.l.], 2007. p. 1–2.

GODERIS, A. *et al.* Discovering scientific workflows: The myexperiment benchmarks. University of Southampton, p. 1–12, 2008.

GOECKS, J. *et al.* Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. **Genome Biol**, v. 11, n. 8, p. 86, 2010.

GROTH, P. *et al.* An architecture for provenance systems. University of Southampton, 2006.

GROUP, D. P. W. *et al.* **ProvONE: A PROV extension data model for scientific workflow provenance**. 2014. Disponível em: <http://vcvcomputing.com/provone-/provone.html>.

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? **International journal of human-computer studies**, Elsevier, v. 43, n. 5, p. 907–928, 1995.

GÜNTHER, C. W.; AALST, W. M. van der. A generic import framework for process event logs. In: SPRINGER. **Business Process Management Workshops**. [S.l.], 2006. p. 81–92.

HASAN, R.; SION, R.; WINSLETT, M. Introducing secure provenance: problems and challenges. In: ACM. **Proceedings of the 2007 ACM workshop on Storage security and survivability**. [S.l.], 2007. p. 13–18.

HENNING, V.; REICHELT, J. Mendeley-a last. fm for research? In: IEEE. **eScience, 2008. eScience'08. IEEE Fourth International Conference on**. [S.l.], 2008. p. 327–328.

HEY, A. J.; TREFETHEN, A. E. The data deluge: An e-science perspective. Wiley and Sons, 2003.

HOLL, S. *et al.* A new optimization phase for scientific workflow management systems. **Future generation computer systems**, Elsevier, v. 36, p. 352–362, 2014.

HOLLINGSWORTH, D. *et al.* The workflow reference model: 10 years on. In: CITESEER. **Fujitsu Services, UK; Technical Committee Chair of WfMC**. [S.l.], 2004.

INGOWASSINK, H. R.; VET, P. der. E-bioflow: Different perspectives on scientific workflows. In: SPRINGER SCIENCE & BUSINESS MEDIA. **Bioinformatics Research and Development: Second International Conference, BIRD 2008, Vienna, Austria, July 7-9, 2008 Proceedings**. [S.l.], 2008. v. 13, p. 243.

JUNG, J.-Y.; BAE, J. Workflow clustering method based on process similarity. In: **Computational Science and Its Applications-ICCSA 2006**. [S.l.]: Springer, 2006. p. 379–389.

JURISTO, N.; MORENO, A. M. **Basics of software engineering experimentation**. [S.l.]: Springer Science & Business Media, 2013.

KACSUK, P. P-grade portal family for grid infrastructures. **Concurrency and Computation: Practice and Experience**, Wiley Online Library, v. 23, n. 3, p. 235–245, 2011.

KAIL, E. *et al.* Dynamic workflow support in guse. In: **Information and Communication, Technology Electronics and Microelectronics (MIPRO),2014 37th International Convention on**. [S.l.: s.n.], 2014. p. 354–359.

KITCHENHAM, B. *et al.* Systematic literature reviews in software engineering–a systematic literature review. **Information and software technology**, Elsevier, v. 51, n. 1, p. 7–15, 2009.

LAENDER, A. H.; GONÇALVES, M. A.; ROBERTO, P. A. Bdbcomp: building a digital library for the brazilian computer science community. In: ACM. **Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries**. [S.l.], 2004. p. 23–24.

LEBO, T. *et al.* Prov-o: The prov ontology. **W3C Recommendation**, v. 30, 2013.

LEHMAN, M. Software's future: Managing evolution. **IEEE software**, IEEE, n. 1, p. 40–44, 1998.

LEHMAN, M. M. Programs, life cycles, and laws of software evolution. **Proceedings of the IEEE**, IEEE, v. 68, n. 9, p. 1060–1076, 1980.

LEHMAN, M. M. Laws of software evolution revisited. In: **Software process technology**. [S.l.]: Springer, 1996. p. 108–124.

LIM, C. *et al.* Prospective and retrospective provenance collection in scientific workflow environments. In: IEEE. **Services Computing (SCC), 2010 IEEE International Conference on**. [S.l.], 2010. p. 449–456.

MAMONE, S. **The IEEE standard for software maintenance**. [S.l.]: ACM, 1994. 75–76 p.

MARINHO, A. *et al.* Provmanager: a provenance management system for scientific workflows. **Concurrency and Computation: Practice and Experience**, Wiley Online Library, v. 24, n. 13, p. 1513–1530, 2012.

MARINHO, A.; WERNER, C. M. L.; MURTA, L. G. P. Provmanager: uma abordagem para gerenciamento de proveniência de workflows científicos. In: **Workshop de Teses e Dissertações em Engenharia de Software, XXIII SBES**. [S.l.: s.n.], 2009. v. 14.

MARSHAK, R. T. Workflow: Applying automation to group processes. In: PRENTICE HALL INTERNATIONAL (UK) LTD. **Groupware**. [S.l.], 1995. p. 71–97.

MATES, P. *et al.* Crowdlabs: Social analysis and visualization for the sciences. In: SPRINGER. **Scientific and Statistical Database Management**. [S.l.], 2011. p. 555–564.

MATTOSO, M. *et al.* Desafios no apoio à composição de experimentos científicos em larga escala. **Seminário Integrado de Software e Hardware, SEMISH**, v. 9, p. 36, 2009.

MCPHILLIPS, T. *et al.* Scientific workflow design for mere mortals. **Future Generation Computer Systems**, Elsevier, v. 25, n. 5, p. 541–551, 2009.

MEDEIROS, C. B. *et al.* Woodss and the web: annotating and reusing scientific workflows. **ACM SIGMOD Record**, ACM, v. 34, n. 3, p. 18–23, 2005.

MENDES, M. **Como organizar uma prova de conceito arquitetural**. 2014. Disponível em: <http://arquiteturasistemas.wordpress.com/2014/01/21/como-organizar-uma-prova-de-conceito-arquitetural/>.

MIRANDA, G. *et al.* Collabcumulus: Uma ferramenta de apoio à análise colaborativa de proveniência em workflows científicos. **SBSC**, 2014.

MISSIER, P. *et al.* D-prov: Extending the prov provenance model with workflow structure. In: **5th USENIX Workshop on the Theory and Practice of Provenance (TaPP 13)**. [S.l.: s.n.], 2013.

MOREAU, L. *et al.* The open provenance model core specification (v1. 1). **Future generation computer systems**, Elsevier, v. 27, n. 6, p. 743–756, 2011.

MOREAU, L.; FOSTER, I. Provenance and annotation of data: International provenance and annotation workshop, ipaw 2006, chicago, il, usa, may 3-5, 2006, revised selected papers. In: . [S.l.]: Springer Science & Business Media, 2006. v. 4145.

MOREAU, L.; MISSIER, P. Prov-dm: The prov data model. World Wide Web Consortium, 2013.

NARDI, A. R. **Uma arquitetura de baixo acoplamento para execução de padrões de controle de fluxo em grades**. Tese (Doutorado) — Universidade de São Paulo, 2009.

NOVAIS, R.; JÚNIOR, P. R. S.; MENDONÇA, M. Timeline matrix: an on demand view for software evolution analysis. In: **Software Visualization (WBVS), 2012 2nd Brazilian Workshop on**. [S.l.: s.n.], 2012. p. 1–8.

OGASAWARA, E. *et al.* Linhas de experimento: Reutilização e gerência de configuração em workflows científicos. In: **2 Workshop E-Science**. [S.l.: s.n.], 2008. p. 31–40.

OINN, T. *et al.* Taverna/mygrid: aligning a workflow system with the life sciences community. In: **Workflows for e-Science**. [S.l.]: Springer, 2007. p. 300–319.

OLIVEIRA, F. T. D. *et al.* Using provenance to improve workflow design. In: **Provenance and Annotation of Data and Processes**. [S.l.]: Springer, 2008. p. 136–143.

OLSON, G. M. The next generation of science collaboratories. In: IEEE. **Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on**. [S.l.], 2009. p. xv–xvi.

PACHECO, C. B. **Reuso e modificação em sistemas de workflow: teoria e estudo de casos**. Dissertação (Mestrado) — Unicamp, 2004.

PEREIRA, W. M.; ARAÚJO, M. A. P.; TRAVASSOS, G. H. Apoio na concepção de workflow científico abstrato para estudos in virtuo e in silico em engenharia de software. In: CITESEER. **Proceedings of 6th Experimental Software Engineering Latin American Workshop (ESELAW 2009)**. [S.l.], 2009. p. 22.

PONCIN, W.; SEREBRENIK, A.; BRAND, M. van den. Process mining software repositories. In: IEEE. **Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on**. [S.l.], 2011. p. 5–14.

ROURE, D. D.; GOBLE, C.; STEVENS, R. The design and realisation of the virtual research environment for social sharing of workflows. **Future Generation Computer Systems**, Elsevier, v. 25, n. 5, p. 561–567, 2009.

SANTOS, E. *et al.* A first study on clustering collections of workflow graphs. Springer, p. 160–173, 2008.

SANTOS, E. *et al.* Vismashup: Streamlining the creation of custom visualization applications. **Visualization and Computer Graphics, IEEE Transactions on**, IEEE, v. 15, n. 6, p. 1539–1546, 2009.

SATLER, M. Utilidad de la tecnología de workflow en los psees. **Dep. de Informática, Universidad de Castilla-La Mancha**, 2004.

SCHMIDT, M.; GLOETZNER, T. Constructing difference tools for models using the sidiff framework. In: ACM. **Companion of the 30th international conference on Software engineering**. [S.l.], 2008. p. 947–948.

SEO, J. *et al.* Retrieving functionally similar bioinformatics workflows using tf-idf filtering. **IPSJ Digital Courier**, Information Processing Society of Japan, v. 3, p. 164–173, 2007.

SHIRASUNA, S.; GANNON, D. Xbaya: A graphical workflow composer for the web services architecture. **Indiana University**, 2006.

SILVA, F. Q. da *et al.* A critical appraisal of systematic reviews in software engineering from the perspective of the research questions asked in the reviews. In: ACM. **Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement**. [S.l.], 2010. p. 33.

SILVA, L. M. da; BRAGA, R.; CAMPOS, F. Composer-science: a semantic service based framework for workflow composition in e-science projects. **Information Sciences**, Elsevier, v. 186, n. 1, p. 186–208, 2012.

SILVA, R. A. C.; SOARES, L. S.; BRAGA, J. L. Workflow aplicado a engenharia de software baseada em processos: uma visão geral. **INFOCOMP Journal of Computer Science**, v. 5, n. 3, p. 76–84, 2006.

SILVA, V. *et al.* Simiflow: Uma arquitetura para agrupamento de workflows por similaridade. **IV e-Science**, p. 1–8, 2010.

SIMMHAN, Y. L. *et al.* Performance evaluation of the karma provenance framework for scientific workflows. In: **Provenance and Annotation of Data**. [S.l.]: Springer, 2006. p. 222–236.

SIQUEIRA, F. *et al.* Como elaborar projetos de pesquisa: linguagem e método. **Rio de Janeiro**, 2008.

SOMMERVILLE, I. Engenharia de software, 9ª edição, tradução: Ivan bosnic e kalinha g. de o. gonçalves. **São Paulo: Person Prentice Hall**, 2011.

STEINMACHER, I.; CHAVES, A. P.; GEROSA, M. A. Awareness support in distributed software development: A systematic review and mapping of the literature. **Computer Supported Cooperative Work (CSCW)**, Springer, v. 22, n. 2-3, p. 113–158, 2013.

STEVENS, R. D.; ROBINSON, A. J.; GOBLE, C. A. mygrid: personalised bioinformatics on the information grid. **Bioinformatics**, Oxford Univ Press, v. 19, n. suppl 1, p. i302–i304, 2003.

TAVARES, J. F. *et al.* Giveme views: uma ferramenta de suporte a evolução de software baseada na análise de dados históricos. **XI Simpósio Brasileiro de Sistemas de Informação (SBSI)**, p. 55–62, 2015.

TAYLOR, I. *et al.* The triana workflow environment: Architecture and applications. In: **Workflows for e-Science**. [S.l.]: Springer, 2007. p. 320–339.

TAYLOR, I. J. *et al.* **Workflows for e-Science: scientific workflows for grids**. [S.l.]: Springer Publishing Company, Incorporated, 2014.

TRAVASSOS, G. H. *et al.* An environment to support large scale experimentation in software engineering. In: IEEE. **Engineering of Complex Computer Systems, 2008. ICECCS 2008. 13th IEEE International Conference on**. [S.l.], 2008. p. 193–202.

VENN, J. I. on the diagrammatic and mechanical representation of propositions and reasonings. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, Taylor & Francis, v. 10, n. 59, p. 1–18, 1880.

VERDI, K. K.; ELLIS, H. J.; GRYK, M. R. Conceptual-level workflow modeling of scientific experiments using nmr as a case study. **BMC bioinformatics**, BioMed Central Ltd, v. 8, n. 1, p. 31, 2007.

VOEGELE, C. *et al.* A laboratory information management system (lims) for a high throughput genetic platform aimed at candidate gene mutation screening. **Bioinformatics**, Oxford Univ Press, v. 23, n. 18, p. 2504–2506, 2007.

WANGENHEIM, C. G. von; RUHE, G. Análise de custo e beneficio de mensuração baseada em gqm-um estudo de caso replicado. Springer Science & Business Media, p. 12, 1999.

WHITGIFT, D. **Methods and tools for software configuration management**. [S.l.]: J. Wiley, 1991.

WOHLIN, C. *et al.* **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.

YU, C.; HUANG, L. A web service qos prediction approach based on time-and location-aware collaborative filtering. **Service Oriented Computing and Applications**, Springer, v. 10, n. 2, p. 135–149, 2016.

YU, L. **A developer's guide to the semantic Web**. [S.l.]: Springer Science & Business Media, 2011.

ZHAO, J. *et al.* Semantically linking and browsing provenance logs for e-science. In: **Semantics of a Networked World. Semantics for Grid Databases**. [S.l.]: Springer, 2004. p. 158–176.

Apêndice A -

Tabela A.1: Descrição do schema do banco de dados.

Tabela	Descrição
Entity	Armazena os dados das instituições que utilizam o E-SECO
Agent	Armazena os dados dos pesquisadores
Activity	Armazena os dados das atividades que compõem os *workflows*
ResearchGroup	Armazena os dados dos grupos de pesquisas
Experiment	Armazena os dados sobre os experimentos
Workflow	Armazena os dados sobre os *workflows*
SGWFC	Armazena informações sobre o SGWfC utilizado pelos *workflows*
Task	Armazena informações sobre as tarefas que compõem os *workflows*
InputPort	Armazena informações sobre as portas de entrada das tarefas que compõem os *workflows*
OutputPort	Armazena informações sobre as portas de saída das tarefas que compõem os *workflows*
IsPartOf	Armazena as associações de pesquisadores com grupos de pesquisa
WasGeneratedBy	Registra quais experimentos foram criados por quais grupos de pesquisas
WasControledBy	Registra quais pesquisadores controlam quais atividades do experimento
WasInformedBy	Armazena o histórico de informações entre as tarefas de um *workflow* e as atividades de um experimento
WasStartedBy	Registra qual tarefa do *workflow* iniciou qual atividade do experimento
WasEndedBy	Registra qual tarefa do *workflow* finalizou qual atividade do experimento
WasAssociatedWith	Registra quais *workflows* estão associados com quais experimentos
Used	Armazenas as informações sobre quais tarefas compõem quais *workflows*
ActedOnBehalfOf	Armazena a ligação entre as portas de comunicação das tarefas que compõem o *workflow*, registrando o fluxo de dados dentro do *workflow*
WasDerivedFrom	Armazena o histórico de derivações que o *workflow* sofreu ao longo do ciclo de vida
WasRevisionOf	Armazena o histórico de revisões que o *workflow* sofreu ao longo do ciclo de vida
WasStartedByWT	Registra qual tarefa inicia o *workflow*
WasEndedByWT	Registra qual tarefa finaliza o *workflow*