



## PROVENIÊNCIA DE DADOS E GESTÃO DE CONFIGURAÇÃO EM PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

DOI: 10.37702/2175-957X.COBENGE.2024.5230

**Autores:** MARCELA GOMES PINHEIRO, RAFAEL DE OLIVEIRA LACERDA, GABRIEL MEDEIROS MACEDO, LAIZA MOTA DE SOUZA, GABRIELLA CASTRO BARBOSA COSTA DALPRA

**Resumo:** Desenvolver softwares com mais qualidade e eficiência tem se tornado cada vez mais desafiador. Nesse cenário, muitas organizações e projetos possuem mecanismos de coleta e análise de dados insuficientes ou mal organizados que resultam em mecanismos limitados, imprecisos, ou que fornecem feedbacks inadequados para gerentes e desenvolvedores. Além do uso de práticas baseadas em dados, o processo de software utilizado é um fator crítico para o desenvolvimento de produtos de software de qualidade. Uma proposta para auxiliar a prática de compartilhamento de dados, é através do uso de modelos de proveniência de dados que garantam o um compartilhamento de dados de processos de software em ambientes distribuídos, como é o caso do PROV-SwProcess. Este trabalho apresenta uma ferramenta em desenvolvimento denominada PROV-SwSystem, capaz de coletar dados, para que posteriormente possam ser analisados e auxiliarem no processo de tomada de decisões, além de realizar a gestão de configuração dos artefatos gerados.

**Palavras-chave:** Proveniência de Dados, Gestão de Configuração, Processos de Desenvolvimento de Software

# PROVENIÊNCIA DE DADOS E GESTÃO DE CONFIGURAÇÃO EM PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

## 1 INTRODUÇÃO

Para o desenvolvimento de softwares de boa qualidade, se faz necessário o alinhamento de dois pilares: boas práticas de Engenharia de Software com um processo de desenvolvimento eficiente. Dessa forma, uma metodologia de desenvolvimento de software possui como objetivo criar sistemas seguros e de qualidade, utilizando estratégias e ações que garantirão que os critérios fundamentais para criação sejam atendidos (PRESSMAN, 2021).

Com o passar dos anos, as práticas de desenvolvimento de software evoluíram e as empresas que atuam neste ramo começaram a adotar práticas orientadas a dados como parte de seus projetos (JANSEN, 2020). Com isso, a utilização de conceitos como o de proveniência de dados se torna relevante, pois, a coleta de dados de forma insuficiente ou mal organizada, pode gerar análises incompletas ou imprecisas (BASILI et al., 2007). A proveniência de dados se refere a um registro histórico que permite a interpretação e a análise sobre os processos ocorridos desde a criação do dado até o seu estado atual, com suas transformações e diagnóstico de problemas que ocorreram durante o ciclo de vida desse dado (LIM et al., 2010).

Uma solução proposta para compartilhar e analisar dados do desenvolvimento de software é o uso de modelos adequados para a captura e compartilhamento de proveniência de dados de processos de software em ambientes distribuídos, como, por exemplo, a internet. Como exemplo de modelo de proveniência de dados, pode-se citar o PROV-SwProcess<sup>1</sup> (COSTA et al., 2021). Esse modelo é uma extensão do modelo PROV<sup>2</sup>, e é capaz de capturar os dados de proveniência de processos de software, além de apoiar a tomada de decisão por parte dos gestores destes processos.

Assim como definido por PRESSMAN e MAXIN (2021), durante as fases do ciclo de vida de um software, não importa qual seja, o sistema necessitará de mudanças e isso altera todo o ciclo de vida definido anteriormente. Desse modo, uma forma de apoiar o desenvolvimento de softwares seguros, é através da aplicação de técnicas de gestão de configuração de software (GCS), que visam o controle e organização das modificações realizadas, buscando manter um histórico de versões antigas (MALHONE et al., 2021). Nesse viés, a GCS pode ser definida como um processo da Engenharia de Software para gerar um estabelecimento consistente dos atributos de um produto de software durante sua vida (BUCHANAN, 2024). Além disso, é uma forma de manter a rastreabilidade do software, bem como a garantia da qualidade do sistema e em todo seu desenvolvimento (MALHONE et al., 2021).

Em resumo, a proveniência de dados está ligada à descrição da origem de um dado e ao processo pelo qual ele passou, enquanto a gestão de configuração visa manter e supervisionar a evolução de um software através do controle formal de versão e solicitação de mudanças. Dessa forma, apesar de possuírem definições diferentes, ambas possuem um objetivo em comum: aumentar a transparência, integridade, rastreabilidade e confiabilidade dos dados e/ou dos sistemas. Diante desse contexto, o presente trabalho apresenta a inserção dos conceitos de GCS em uma ferramenta em desenvolvimento

<sup>1</sup> <https://prov.linceonline.com.br/modelo/index.html>

<sup>2</sup> <https://www.w3.org/TR/prov-primer/>

para capturar, armazenar e analisar dados de processos de software, conforme especificado pelo modelo PROV-SwProcess. Além disso, cabe ressaltar que a ferramenta apresentada nesse trabalho é capaz de auxiliar a execução de disciplinas práticas nas áreas de Engenharia de Software que envolvem o desenvolvimento de sistemas e/ou utilizam da metodologia PBL (*Project-Based Learning* - Aprendizagem Baseada em Projetos), visto que, através da proposta apresentada no presente trabalho, é possível armazenar e gerenciar as mudanças que ocorrem nos sistemas desenvolvidos em sala, bem como os artefatos gerados. O restante deste trabalho está estruturado da seguinte forma: na **Seção 2** têm-se os trabalhos relacionados a esta proposta; a **Seção 3** descreve a proposta da ferramenta PROV-SwSystem; a **Seção 4** apresenta os resultados preliminares; por fim, a **Seção 5** apresenta as considerações finais, seguida pelas referências bibliográficas utilizadas.

## 2 TRABALHOS RELACIONADOS

Em busca de obter uma base teórica dos trabalhos em desenvolvimento na área, bem como os principais desafios apontados pela literatura, foram analisados os trabalhos descritos a seguir.

No trabalho intitulado “Configuration Management process implementation of ITIL framework” (ESTEVES e ALVES, 2013), é apresentado o resultado de um estudo sobre a inserção da técnica de Gestão de Configuração (GC) na Secretaria Regional da Cultura, Turismo e Transportes (SRT), órgão público da Região Autónoma da Madeira (RAM). Como uma forma para coletar dados, neste caso, quais as técnicas de GC que poderiam estar sendo executadas, foram utilizados questionários. Como resultado, foi possível entender quais as funcionalidades deveriam ser implementadas, seguindo as recomendações do framework ITIL e como o processo de GC poderia trazer melhorias para SRT. Porém, ao final do estudo, não foi possível implementar completamente essa técnica (abordando todas as atividades necessárias), visto que para que obtivesse sucesso acarretaria uma grande mudança organizacional, o que afetaria os procedimentos e funções realizados habitualmente.

Em “DKD-S: An Ontology-based Tool for Global Software Development” (ROCHA et al., 2021), é apresentada uma ferramenta denominada DKDs, uma ferramenta que utiliza uma ontologia denominada DKDOnto (ROCHA et al., 2018). Essa ferramenta permite consultas e atualizações na ontologia, além do compartilhamento de informações em cenários de desenvolvimento distribuído. Recomendada para gestores e líderes de equipes técnicas, possui 4 módulos, sendo eles: **inferência**: capaz de deduzir informações sobre a DKOnto; **consultas**: módulo responsável pelas consultas realizadas pelos usuários; **views**: que garante acesso aos relatórios desenvolvidos, conforme a necessidade de cada usuário; e **gerenciamento**: onde é possível acesso direto a ontologia, para inserção, remoção e edição dos dados. Apesar de apresentarem resultados, a ontologia descrita, bem como a ferramenta, não foram validadas em um ambiente real, o que é considerado um aspecto limitante. Além disso, não é mencionado sobre técnicas de GCS.

No trabalho “Git2PROV: Exposing Version Control System Content as W3C PROV” (DE NIES et al., 2013), é descrita uma ferramenta capaz de apresentar dados de proveniência armazenados em sistemas de controle de versão (SCV), como o Git, seguindo os padrões do W3C PROV (GROTH; MOREAU, 2013). Dessa forma, é possível publicar essa proveniência coletada dos SCV e permitir que outras ferramentas habilitadas para o PROV possam utilizar esses dados já coletados.

Em síntese, os trabalhos mencionados anteriormente propõem uma ferramenta para apoiar o desenvolvimento de software em cenário distribuído, com base na ontologia DKDOnto, tratam de gestão de configuração com uso de *frameworks* ou utilizam de uma ferramenta para capturar os dados de proveniência de repositórios Git. No entanto, não é mencionada uma ferramenta capaz de realizar as duas coisas simultaneamente. Tendo em vista que um desafio existente é a organização e a manipulação dos dados do processo de desenvolvimento de software para que os mesmos atendam um modelo de proveniência (que no caso do presente trabalho será utilizado o PROV-SwProcess), propõem-se uma ferramenta que realiza a captura e a manipulação desses dados atendendo às especificações do modelo citado, além da inclusão da GCS.

### 3 PROPOSTA

Como uma forma de auxiliar no processo de captura e manipulação de dados de proveniência seguindo as especificações do modelo PROV-SwProcess, incluindo a GCS, propõe-se uma ferramenta, em desenvolvimento em um projeto de Iniciação Científica por alunos do curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), campus Leopoldina, denominada PROV-SwSystem. Essa ferramenta, além de realizar a captura, armazenamento e análise de dados de processos de software, possui de forma integrada conexão com o GitHub, como ferramenta de apoio à GCS. Para o desenvolvimento do presente trabalho foram definidas 5 etapas a serem realizadas de forma sistemática: **ETAPA 1:** Revisão bibliográfica; **ETAPA 2:** Elicitação de requisitos e modelagem do sistema; **ETAPA 3:** Desenvolvimento do sistema; **ETAPA 4:** Integração do sistema com a ferramenta de apoio à GCS **ETAPA 5:** Testes e validação do sistema.

Após a conclusão da **ETAPA 1:** Revisão bibliográfica, foi iniciada a **ETAPA 2**, tendo como objetivo a elicitação dos requisitos. O resultado obtido foi um protótipo de média fidelidade<sup>3</sup>. Além disso, foi desenvolvido o diagrama de casos de uso com as funcionalidades do software e os atores envolvidos nas possíveis utilizações das mesmas. Ao todo, foram definidos 3 atores distintos: I) Pessoa, II) Organização, III) Equipe e 22 casos de uso. Após a etapa de elicitação de requisitos e modelagem do sistema, seguiu-se para **ETAPA 3:** Desenvolvimento do sistema. Para o *front-end*, foram escolhidos o HTML5 como forma de estruturação do conteúdo, juntamente com o CSS3 e o Bootstrap 5 para estilização das páginas. Além disso, foram utilizados JavaScript e a biblioteca jQuery para comunicação com servidores e para gerir eventos nas interações dos usuários. Já para o desenvolvimento do *back-end*, o Node.js foi escolhido juntamente com a estrutura Express, visto que o mesmo possui diversas funcionalidades disponibilizadas para estruturação de sites. Além disso, para interagir com a base de dados MySQL, foi utilizada a biblioteca “mysql2” que suporta consultas SQL e interações assíncronas. As rotas da API foram estruturadas em arquivos seguindo os princípios da arquitetura MVC (*Model View Controller*) para garantir a separação das tarefas e a organização do código. A **ETAPA 3** ainda segue em desenvolvimento, visto que a ferramenta não foi finalizada, de forma paralela à **ETAPA 4**. Para essa quarta etapa foi escolhido o GitHub como um sistema de controle de versões e como forma de realizar a conexão com o PROV-SwSystem foi utilizada a “*GitHub REST API*”, utilizada para as interações dos recursos disponibilizados pelo GitHub, como *commits*, e a biblioteca

<sup>3</sup> O protótipo pode ser visualizado em:

<https://www.figma.com/proto/2R9IDFJQMTLlua3PQ2xck52/novo?node-id=45-1593&t=2Pd4O7TiF6dDHH2-1&scaling=min-zoom&page-id=1%3A616>

Octokit, para realizar as requisições e chamadas de ação da API. **ETAPA 5:** A etapa de testes e validação ainda não foi realizada.

#### 4 RESULTADOS PRELIMINARES

Como resultado preliminar do trabalho tem-se uma ferramenta capaz de capturar dados de processos de software, segundo as especificações do modelo PROV-SwProcess, que ainda se encontra em desenvolvimento, porém já consta com a implementação do processo de gestão de configuração e controle de versões. Esse sistema será útil em situações onde não são utilizadas ferramentas destinadas especificamente para captação desses tipos de dados, seja na indústria, seja no meio acadêmico. Dessa maneira, posteriormente os dados poderão ser manipulados e analisados de forma a auxiliar na melhoria contínua dos processos e otimizar as tomadas de decisão.

O PROV-SwSystem já possui a maior parte de suas páginas já desenvolvidas (Figura 1) completamente, ou seja, tanto o *front-end* quanto o *back-end* já se encontram em seu estágio final de funcionamento.

Figura 1 - Exemplo de tela do sistema em desenvolvimento PROV-SwSystem



The screenshot displays the 'Procedimentos' (Procedures) page of the PROV-SwSystem. The interface includes a top navigation bar with the 'PSW' logo and a user profile 'Fulano'. A left sidebar lists menu items: Artefatos, Procedimentos, Recursos, Stakeholders, Atividades, and Processos. The main content area features a form with the following fields: 'Tipo' (text input), 'Título' (text input), 'Descrição' (text area), 'Upload do Arquivo' (with 'Escolher arquivo' and 'Nenhum arquivo escolhido' buttons), and 'Versão Revisada' (checkbox). At the bottom of the form are 'Cancelar' and 'Cadastrar' buttons. A search bar with a 'Pesquisar' button is located at the top right of the main content area.

Com a ferramenta proposta é possível capturar e armazenar dados de execução de processos de software, o que inclui: Os processos executados, envolvendo a identificação dos responsáveis e suas correspondentes ações, além de documentar as atividades específicas realizadas em cada processo, incluindo horários de início e fim; Como componentes obrigatórios, destacam-se: os *stakeholders* associados às suas atividades e com seus papéis específicos, bem como quaisquer artefatos alterados, criados ou utilizados durante as atividades; Já como componentes opcionais podem ser citados: os procedimentos implementados para a realização das atividades, os recursos de hardware ou software empregados, além das responsabilidades existentes entre os *stakeholders*; além disso, qual o modelo padrão de processo e qual o pretendido, a fim de permitir a captura e análise da proveniência retrospectiva do processo.

No que diz respeito à técnica de GCS, durante a utilização do sistema, o usuário precisa realizar previamente o cadastro na plataforma GitHub, bem como criar um

repositório e uma chave de acesso (*token*) e inserir essas informações no PROV-SwSystem conforme exibido na Figura 2.

Figura 2 - Exemplo de tela PROV-SwSystem gestão de configuração e controle de versão



The screenshot shows the 'Processos de Software' page in the PROV-SwSystem interface. The page has a blue header with the 'PSW' logo and a user profile 'Fulano'. A left sidebar contains a navigation menu with items like 'Artefatos', 'Procedimentos', 'Recursos', 'Stakeholders', 'Atividades', and 'Processos'. The main content area is titled 'Processos de Software' and includes a search bar and a 'Listar Todos' button. Below this, there are three input fields for 'Usuário do github:', 'Nome do repositório:', and 'Token de acesso:', followed by a 'Conectar' button. Further down, there are fields for 'Nome:' and 'Descrição:'. At the bottom, there is a section for 'Stakeholders Atribuídos' with a 'Stakeholder:' label, a plus sign, and a 'Cadastrar' button. At the very bottom of the form area, there are 'Cancelar' and 'Cadastrar' buttons.

Após a inserção dessa nova função na página, é possível que os usuários consigam, de forma automática, inserir novos artefatos no repositório do GitHub cadastrado, além de manter um controle do trabalho que está sendo desenvolvido por outros *stakeholders* envolvidos no projeto de forma simultânea.

## 5 CONSIDERAÇÕES FINAIS

Desenvolver softwares com mais qualidade e eficiência tem se tornado cada vez mais desafiador. Nesse cenário, muitas organizações e projetos possuem mecanismos de coleta e análise de dados insuficientes ou mal organizados, resultando em mecanismos limitados, imprecisos, ou que fornecem *feedbacks* inadequados para gerentes e desenvolvedores. Além do uso de práticas baseadas em dados, o processo de software utilizado é um fator crítico para o desenvolvimento de produtos de software de qualidade e “os dados do processo de engenharia de software são uma fonte valiosa de informações sobre a história e evolução de um projeto de software” (BACHMANN e BERNSTEIN, 2009).

As práticas baseadas em dados têm desafios que envolvem um ciclo de atividades para preparar os dados a serem usados. Este ciclo inclui estratégias para coleta, armazenamento, representação, integração, compartilhamento e manutenção de dados. Pensando nisso, uma forma de auxiliar a produção de software seria a coleta dos dados de proveniência do processo de desenvolvimento de software mediante um modelo de representação de dados para, em seguida, realizar um estudo e uma análise sobre esses dados pensando em otimizar tomadas de decisão.

A principal contribuição do presente trabalho consiste em apresentar o PROV-SwSystem, uma ferramenta em desenvolvimento capaz de coletar dados de proveniência de processos de software, através das especificações descritas pelo modelo PROV-SwProcess e realizar a gestão de configuração. A captura desses dados permitirá

que, posteriormente, eles possam ser analisados a fim de otimizar o processo de tomada de decisão sobre os processos de desenvolvimento. Como trabalhos futuros, propõe-se a finalização da ferramenta, testes em ambientes reais e a correção de possíveis erros de experiência do usuário.

## AGRADECIMENTOS

Agradecemos ao CEFET-MG, ao Laboratório de Iniciação Científica e Extensão da Computação (LINCE) e ao CNPq pelo apoio para a realização deste trabalho.

## REFERÊNCIAS

BACHMANN, A., BERNSTEIN, A. Software process data quality and characteristics: A historical view on open and closed source projects. Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops. Anais...New York, NY, USA: ACM, 2009.

BASILI, V., ROMBACH, D., SCHNEIDER, K., KITCHENHAM, B., PFAHL, D., SELBY, R. (Eds.). Empirical Software Engineering Issues. Critical Assessment and Future Directions: International Workshop, Dagstuhl Castle, Germany, June 26-30, 2006, Revised Papers (Vol. 4336). Springer. 2007.

BUCHANAN, I. Gerenciamento de configuração: definição e benefícios. Disponível em: <<https://www.atlassian.com/br/microservices/microservices-architecture/configuration-management>>. Acesso em: 14 mai. 2024.

COSTA, G. C. B., WERNER, C., BRAGA, R., TEIXEIRA, E. N., STRÖELE, V., ARAÚJO, M. A. P., MIGUEL, M. A. Design, Application and Evaluation of PROV-SwProcess: A PROV extension Data Model for Software Development Processes. Journal of Web Semantics, v. 71, p. 100676, 2021.

DE NIES, T., MAGLIACANE, S., VERBORGH, R., COPPENS, S., GROTH, P., MANNENS, E., VAN DE WALLE, R. Git2PROV: Exposing Version Control System Content as W3C PROV. Published in 2013 in Poster and Demo Proceedings of the 12th International Semantic Web Conference., v. 1035, p. 125–128, 2013.

ESTEVES, R., ALVES, P. Implementação do processo gestão da configuração da framework ITIL. In: 8th Iberian Conference on Information Systems and Technologies, CISTI 2013. 2013.

GROTH, P., MOREAU, L. PROV-Overview: An Overview of the PROV Family of Documents. W3C Working Group Note, 2013.

JANSEN, S. A focus area maturity model for software ecosystem governance, Information and Software Technology, vol. 118, ISSN 0950-5849, 2020.

LIM, C., LU, S., CHEBOTKO, A., FOTOUHI, F. Prospective and retrospective provenance collection in scientific workflow environments. 2010 IEEE International Conference on Services Computing. Anais...IEEE, 2010.

MALHONE, M. M., FRIGERI, M. A importância do gerenciamento de configuração para o ciclo de vida do software: um estudo de caso baseado nas diretrizes da engenharia de

software. RBTI-Revista Brasileira em Tecnologia da Informação, v. 3, p. 1–60, 2021.

PRESSMAN, R. S., MAXIM, B. R. Engenharia de software - 9.ed. [s.l.] McGraw Hill Brasil, 2021.

ROCHA, R., LEANDRO, R., SILVA, I., ARAUJO, J., BION, D., FREITAS, F., CORDEIRO D., GOMES, A., AZEVEDO, R. DKD-S: An ontology-based tool for global software development. 2021 16th Iberian Conference on Information Systems and Technologies (CISTI). Anais...IEEE, 2021.

ROCHA, R., ARAÚJO, A., CORDEIRO, D., XIMENES, A., TEIXEIRA, J., SILVA, G., DA SILVA, D., ESPINHARA, D., FERNANDES, R., AMBROSIO, J., DUARTE, M., AZEVEDO, R. DKDOnto: An ontology to support software development with distributed teams. 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Anais...IEEE, 2018.

## DATA PROVENANCE AND CONFIGURATION MANAGEMENT IN SOFTWARE DEVELOPMENT PROCESSES

**Abstract:** *Developing software with greater quality and efficiency has become increasingly challenging. In this scenario, many organizations and projects have insufficient or poorly organized data collection and analysis mechanisms, resulting in limited, inaccurate, or inadequate feedback for managers and developers. In addition to the use of data-driven practices, the software process used is a critical factor in the development of quality software products. One proposal to assist data-sharing practices is the use of data provenance models that guarantee the sharing of software process data in distributed environments, such as PROV-SwProcess. This work presents a tool under development called PROV-SwSystem, capable of collecting software development process provenance data that can be later analyzed and assist in the decision-making process, as well as performing configuration management of the generated artifacts.*

**Keywords:** *Data Provenance, Configuration Management, Software Development Processes.*

