

# @Minerva\_web: Análise de Colaboração em Processos de Negócio por meio de Dados de Proveniência Multimodais<sup>\*†</sup>

Ana Luisa Esposito, Lucas Rogério Silva Ferreira, Maria Luiza Falci, Daniel de Oliveira

<sup>1</sup>Instituto de Computação – Universidade Federal Fluminense (IC/UFF)  
Niterói – RJ – Brasil

{luisa\_esposito, lucasrogerio, marialuizafalci}@id.uff.br, danielcmo@ic.uff.br

**Resumo.** Neste artigo apresentamos o @Minerva\_web, um sistema web que permite a análise de grafos de colaboração em processos de negócio gerados a partir da extração de dados de proveniência multimodais. A colaboração entre participantes dos processos pode ser vista como um grafo de proveniência, uma vez que agentes (i.e., pessoas) executam tarefas (i.e., atividades) sobre documentos (i.e., entidades). Entretanto, a colaboração nas organizações acontece de diversas formas, e.g., e-mails e vídeo chamadas. Extrair dados de proveniência a partir dessa miríade de tipos de dados (em especial não estruturados) é um desafio. O @Minerva\_web é capaz de identificar entidades nomeadas (e.g., pessoas, processos, etc.) em dados não estruturados automaticamente e representar os dados extraídos como um grafo de proveniência, que pode ser visualizado e consultado pelo usuário. A base de dados segue a recomendação W3C PROV, favorecendo a comparação e explicação do processo de negócio. O @Minerva\_web é uma solução de código aberto e pode ser obtida em <https://github.com/luisaesposito/minerva-backend>.

## 1. Introdução

No cenário corporativo global, tem se tornado cada vez mais importante que as organizações estejam em constante desenvolvimento para se manterem competitivas [Bicheno and Holweg 2016]. Apesar de vários fatores influenciarem na capacidade competitiva de uma organização, um dos fatores que mais impactam é uma boa definição dos processos de negócio da organização. Processos de negócio descrevem as atividades executadas por organizações para gerar um produto ou serviço. Existe uma miríade de ferramentas e notações (e.g., BPMN - *Business Process Management Notation*) para modelar os processos de uma organização. Apesar da modelagem do processo ser fundamental, ela não é uma garantia de sucesso.

Na prática, processos podem estar muito bem modelados, mas não serem seguidos pelas pessoas envolvidas. Dessa forma, a organização deve ser capaz de analisar se um determinado processo está sendo seguido executado foi modelado, e se pode ser melhorado. Para atingir esse objetivo, se torna imprescindível analisar dados históricos sobre as execuções dos processos. Dessa forma, os dados de proveniência [Buneman et al. 2001] podem ser aplicados nesse cenário. Os dados de proveniência descrevem a origem de um

\*Vídeo de demonstração: <https://www.youtube.com/watch?v=HPkSirrtKpE>

†O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq e FAPERJ.

dado, o processo de transformação que levou até esse dado e os atores que atuaram nesse processo. No contexto deste artigo, os dados de proveniência permitem identificar quais pessoas contribuíram para uma determinada atividade ou processo. Esse tipo de informação permite identificar como está sendo realizada a colaboração em processos de negócio, e se o processo modelado está de fato sendo seguido.

Entretanto, diferentemente de outras áreas onde os dados de proveniência se encontram estruturados (*e.g.*, bioinformática [Salazar et al. 2021]), os dados de proveniência de processos de negócio são naturalmente multimodais. As interações entre participantes dos processos nas organizações acontecem de diversas formas, *e.g.*, *e-mails*, vídeo chamadas, *etc.* O problema de termos essa miríade de formas de comunicação é que a maioria da informação sobre a execução dos processos se encontra somente em arquivos de formatos heterogêneos e comumente não estruturados. Assim, é fundamental que possa extrair os dados de proveniência de todas essas fontes de dados no momento da análise da colaboração em tais processos. Apesar de fundamental, essa extração está longe de ser trivial. O processamento de dados de proveniência multimodais inclui a identificação de *Agentes* (*i.e.*, pessoas ou sistemas envolvidos), *Atividades* (*i.e.*, ações executadas) e *Entidades* (*i.e.*, objetos) não somente em dados estruturados, mas também em dados semiestruturados, e principalmente em dados não estruturados (*e.g.*, texto livre). Apesar de existir um padrão definido para representação de dados de proveniência (o W3C PROV [Moreau and Groth 2013]), ele não trata da captura e processamento de dados multimodais, se limitando a representar a proveniência de forma estruturada.

Em trabalhos anteriores, foi proposta a abordagem MINERVA (*Multimodal busI-NEss pRoVenance Analysis*) [Falci et al. 2021, Falci et al. 2020], que tem como objetivo a extração de dados de proveniência multimodais de processos de negócio para analisar a colaboração entre os participantes do processo. A ideia da MINERVA era ser uma arquitetura de referência, e apesar de ter sido mostrada a viabilidade da mesma por meio de um estudo com dados reais de uma empresa de consultoria (*i.e.*, Dheka Consultoria<sup>1</sup>), muitas etapas do processo eram executadas e orquestradas de forma manual. Não havia uma ferramenta que automatizasse todo o fluxo de atividades da MINERVA, o que tornava a execução uma tarefa complexa, repetitiva e pouco eficiente. Neste artigo de demonstração, apresentamos o @Minerva\_web, um sistema *web* para extração de proveniência multimodal e geração do grafo de colaboração do processo de negócio. O @Minerva\_web automatiza todas as etapas descritas na arquitetura de referência MINERVA, mesmo as que envolvem processamentos mais intensivos como as etapas de processamento de linguagem natural. O presente artigo está dividido em três seções além dessa Introdução. A Seção 2 apresenta a arquitetura proposta do @Minerva\_web. A Seção 3 discute a demonstração, e, finalmente, a Seção 4 conclui o artigo.

## 2. Arquitetura do @Minerva\_web

O @Minerva\_web é um sistema *web* baseado na arquitetura de referência MINERVA, e oferece uma interface onde usuários podem carregar os dados de diferentes fontes que são processados de forma automática para a geração do grafo de colaboração. O @Minerva\_web permite a visualização e interação com o grafo de colaboração, além de permitir a submissão de consultas sobre o mesmo, uma vez que persiste o grafo em um Sistema de Gerência de Banco de Dados (SGBD) orientado a grafos (*i.e.*, Neo4J).

---

<sup>1</sup><https://www.dheka.com.br/>

O @Minerva\_web é baseado em arquitetura de microsserviços [Newman 2015] e utiliza contêineres para atender às demandas de cada componente da aplicação de maneira isolada e otimizada. O @Minerva\_web é composto por uma aplicação *front-end*, responsável pela interface com usuário, e três serviços no *back-end* (Figura 1): (i) a *Minerva API*, responsável pelas operações *CRUD* (*Create, Read, Update e Delete*) de metadados de projetos e arquivos, gerenciamento de arquivos e comunicação com o *front-end*; (ii) o *File Processor*, responsável pelo tratamento dos áudios e vídeos, a comunicação com a *API* de transcrição e a execução do *script* de Processamento de Linguagem Natural (PLN); e (iii) o *Graph Manager*, responsável pelas operações *CRUD* do grafo. O armazenamento dos metadados sobre os arquivos que são analisados no @Minerva\_web é realizado em um SGBD relacional (*i.e.*, PostgreSQL), enquanto que a rede de colaboração resultante do processamento dos arquivos fica armazenada em um SGBD orientado a grafos. Toda a comunicação entre os serviços na aplicação acontece via chamadas *HTTP* ou por emissão e captura de mensagens assíncronas. Por serem aplicações Java, a comunicação com o banco de dados relacional acontece via *driver* JDBC e com o Neo4j via *driver* Bolt.

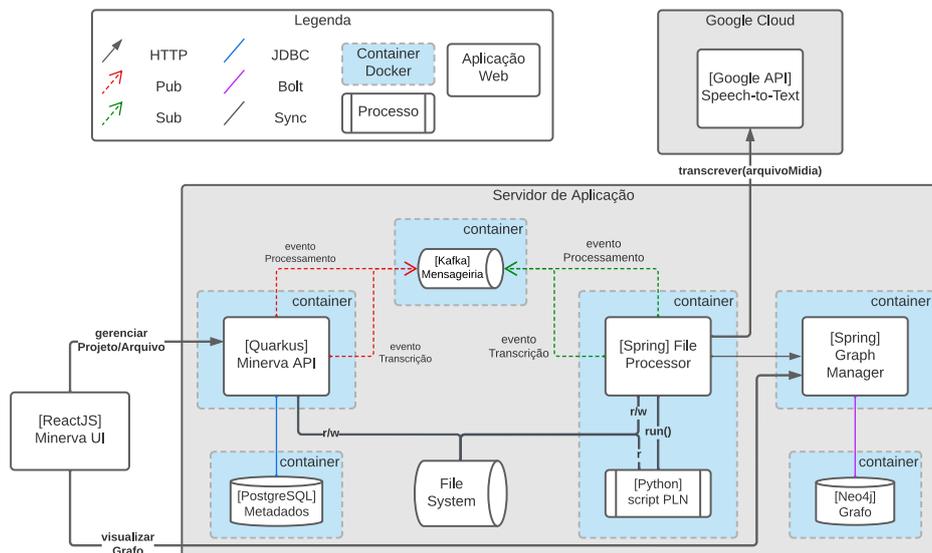


Figura 1. Arquitetura do @Minerva\_web

O processo de execução do @Minerva\_web se inicia com a submissão de um arquivo pelo usuário a partir do *front-end* (Minerva UI). O *front-end* foi desenvolvido utilizando *ReactJS*, uma biblioteca *JavaScript* voltada para criação de componentes de aplicações *web*. O arquivo submetido é enviado ao serviço *Minerva API*. Para desenvolver o serviço *Minerva API* foi utilizado o *framework* *Quarkus*. Dependendo do tipo de arquivo enviado, a API define se é necessário o processamento de linguagem natural ou simplesmente a extração dos dados já estruturados. Uma mensagem é enviada então a um *broker* de mensageria (*i.e.*, *Kafka*) para que seja processado por outro serviço. Além disso, a API registra os metadados dos arquivos recebidos no banco de metadados e grava os arquivos no sistema de arquivos definido (*e.g.*, *HDFS*).

O serviço *File Processor* então consome as mensagens enviadas pela API e orquestra o processamento dos arquivos. O *File Processor* foi implementado usando o *Spring Boot* e invoca um *script* Python para processamento de linguagem natural. O *script* utiliza a bi-

biblioteca de PLN *spaCy*<sup>2</sup>. Com ela, é possível normalizar as entradas de texto e extrair as entidades nomeadas que são utilizadas na etapa de geração do grafo. O *script* possui comportamentos distintos dependendo do tipo de arquivo de entrada. Caso o arquivo processado seja estruturado (*e.g.*, CSV), com cada uma de suas linhas representando um dado de proveniência único (assim como apresentado na Tabela 1), o *script* busca valores nas colunas “Processo”, que indica o processo de negócio, “Organização”, que indica a organização dona do processo, e “Conteúdo”, que indica o conteúdo de onde são filtradas as entidades *Pessoa* e *Organização*.

Se o arquivo não for estruturado, *e.g.* texto livre, seu conteúdo é processado pelo *spaCy* e as entidades são extraídas diretamente dele, enquanto os atributos “Processo” e “Organização” são informados separadamente pelo usuário, passados como parâmetros do *script*. Caso o arquivo seja um áudio, ele é enviado primeiramente para a *API Speech-To-Text*, do Google, e somente depois que o áudio foi transcrito é que o *spaCy* é invocado. Para que a transcrição tenha melhor desempenho, os arquivos de áudio enviados devem estar no formato de compressão *Linear PCM* com taxa de amostragem de 16kHz. O *File Processor* realiza todos os tratamentos necessários antes do envio à *API* utilizando a ferramenta *ffmpeg*. Ao final do processamento, o dados resultantes do processamento são estruturados como um objeto que contém os atributos processo de negócio, o entidade, o tipo de entidade e a organização de origem. Ao fim da execução, é retornado um *JSON* cujo conteúdo é uma lista de tais objetos.

**Tabela 1. Exemplo de dado estruturado de entrada**

Organização	Processo	Conteúdo	...
Org 1	Negócio 3	Lucas entrou em contato com RH	...
Org 2	Negócio 5	Daniel definiu o escopo hoje	...
Org 1	Negócio 7	Reunião com Miguel e Luisa realizada hoje	...
Org 4	Negócio 1	Proposta regularizada pela ANEEL	...
Org 6	Negócio 2	Maria ligou tirando algumas dúvidas	...
...	...	...	...

O arquivo *JSON* resultante da execução do *script* é enviado pelo *File Processor* via *REST client* para o serviço *Graph Manager*. No caso do serviço *Graph Manager* foi utilizado o *Spring Data Neo4j* para o mapeamento de objetos do grafo. O serviço itera sobre a lista de itens recebidos no arquivo *JSON* e persiste o nó *Processo* e sua relação com o nó *Pessoa* ou nó *Organização*, conforme o tipo da entidade do item iterado. A *Organização* de origem é armazenada como um atributo do nó *Pessoa* no grafo.

### 3. Demonstração

O @Minerva\_web possui código aberto e pode ser obtido nos repositórios <https://github.com/luisaesposito/minerva-backend> (*Back-end*) e <https://github.com/luisaesposito/minerva-frontend> (*front-end*). Além disso, se encontra disponível em <https://www.youtube.com/watch?v=HPkS1rrtKpE>, um vídeo de demonstração do sistema. A demonstração do @Minerva\_web seguirá um estudo de caso com dados fictícios (de acordo com a Lei Geral de Proteção aos Dados em vigência no Brasil, somos impedidos de usar dados reais na demonstração). Foram simuladas fontes de dados como gravações de áudio, mensagens em texto e *logs* de eventos que

<sup>2</sup><https://spacy.io/>

contém comentários sobre o processo. O CSV utilizado para o estudo, representado na Tabela 2, segue a estrutura já apresentada anteriormente na Tabela 1. O arquivo de áudio foi gravado pelos autores com a frase “Ricardo informou que está trabalhando no projeto para apresentar à diretoria até o final do mês e que nos informa quando tiver um retorno”. O arquivo de texto submetido simula um *e-mail* com o conteúdo “Boa noite! Eu conversei com o Joseph hoje pela manhã e ele me explicou a alteração do produto feita pelo Salomão na segunda. Att, Ronaldo Lima”. Apesar de propormos esse estudo de caso para a demonstração, encorajamos os participantes do SBBD a submeter seus próprios arquivos para teste no dia do evento.

**Tabela 2. Conteúdo fictício do CSV utilizado na demonstração.**

organizacao	processo	conteudo
Org 1	Proc 1	João alinhou as entregas com Caio na sede da Org 2
Org 1	Proc 3	Luiza terminou os ajustes necessários
Org 1	Proc 2	Pedro se reuniu com João hoje
Org 1	Proc 4	Caio ligou para Jorge
Org 1	Proc 4	Luiza pediu reajuste do contrato para João
Org 2	Proc 6	Contrato enviado por Francisco
Org 2	Proc 7	Aguardando aprovação de Maria
Org 2	Proc 5	Daniel se reuniu hoje com Tiago
Org 2	Proc 8	Maria assinou o contrato da Org 1 com Bianca
Org 2	Proc 8	Bianca se reuniu hoje com Tiago

A demonstração se inicia com a criação de um novo Projeto no @Minerva\_web. Em seguida, são adicionados ao Projeto os arquivos citados anteriormente (Figura 2(A)). Como arquivos de texto e CSV já devem estar pré-processados para remoções de abreviações, *etc*, eles já são considerados como “Aprovados” para processamento assim que enviados. Por sua vez, o texto transcrito do arquivo de áudio precisa ser revisado e aprovado pelo usuário. É importante ressaltar que ao carregar um arquivo de áudio ou texto livre, o usuário deve informar a qual processo e organização o conteúdo daquele arquivo se refere (Figura 2(B)). Uma vez que todos os arquivos tenham sido devidamente revisados e aprovados, o usuário deve selecionar a opção “Processar Arquivos” para gerar o grafo de colaboração. Uma vez gerado, o grafo passa a estar disponível na aba “Grafo” (Figura 2(C)). O @Minerva\_web disponibiliza três consultas já configuradas para o usuário: (i) visualização do grafo completo, (ii) pessoas que participam de vários processos e (iii) processos com múltiplos participantes. Essas consultas são as mesmas propostas originalmente por [Falci et al. 2021].

#### 4. Conclusão

A execução de processos de negócio gera um conjunto rico de dados de proveniência em diferentes formatos. Esses dados podem ser usados para identificar colaborações entre pessoas no processo e, dessa forma, verificar se o processo está sendo executado conforme foi modelado. A maioria das abordagens existentes utiliza apenas *event logs* para realizar análises, o que limita a quantidade e a diversidade de informações que podem ser obtidas. Ao considerarmos dados de proveniência representados em múltiplos formatos, *i.e.*, proveniência multimodal, o grafo de colaboração passa a ser mais rico e a refletir melhor a realidade da organização. Neste artigo de demonstração, apresentamos o @Minerva\_web, um sistema *web* que permite analisar o modo que as pessoas colaboram durante a execução de processos de negócio, a partir da extração de dados de proveniência multimodais. O @Minerva\_web

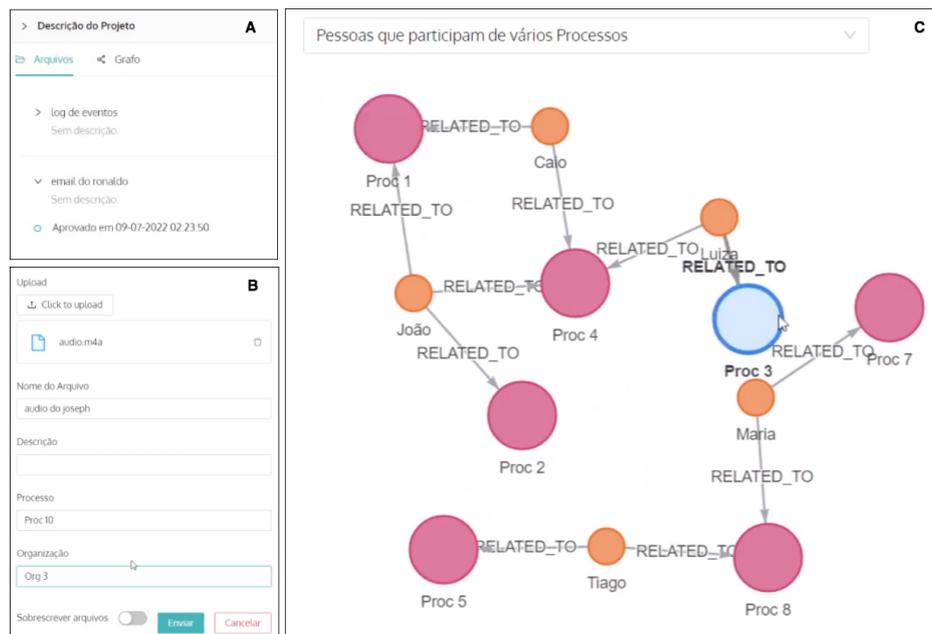


Figura 2. Interface do @Minerva.web

mostrou que é capaz de extrair dados de proveniência multimodal e enriquecer o grafo de colaboração na organização. Trabalhos futuros incluem o processamento de vídeos dentro do @Minerva.web.

## Referências

- [Bicheno and Holweg 2016] Bicheno, J. and Holweg, M. (2016). *The lean toolbox: A handbook for lean transformation*, volume 5. PICSIE books Buckingham, UK.
- [Buneman et al. 2001] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A characterization of data provenance. In *ICDT 2001*, pages 316–330. Springer.
- [Falci et al. 2020] Falci, M. L. F., Magdaleno, A. M., Braganholo, V., Paes, A., and de Oliveira, D. (2020). Análise de colaboração em processos de negócio por meio de sgbds de grafos e dados de proveniência multimodais. In *SBBD 2020*, pages 169–174. SBC.
- [Falci et al. 2021] Falci, M. L. F., Magdaleno, A. M., Paes, A., Braganholo, V., and de Oliveira, D. (2021). Multimodal provenance-based analysis of collaboration in business processes. *J. Inf. Data Manag.*, 12(5).
- [Moreau and Groth 2013] Moreau, L. and Groth, P. (2013). *Provenance: An Introduction to PROV*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers.
- [Newman 2015] Newman, S. (2015). *Building microservices - designing fine-grained systems, 1st Edition*. O’Reilly.
- [Salazar et al. 2021] Salazar, V., Cavalcante, J., de Oliveira, D., Thompson, F. L., and Mattoso, M. (2021). Bioprov - A provenance library for bioinformatics workflows. *J. Open Source Softw.*, 6(67):3622.