RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

ORBITA: Um framework de recomendação de automação baseado em Task Mining e Large Language Models

ORBITA: An automation recommendation framework based on Task Mining and Large Language Models

Márcio da Silva Bastos - Lattes: https://lattes.cnpq.br/4180847255464415; UNIFACCAMP Campo Limpo Paulista, SP, Brasil

Resumo

A transformação digital impulsiona organizações a buscarem eficiência operacional por meio da automação de processos. A Automação Robótica de Processos (RPA) consolidou-se como solução para tarefas repetitivas, porém apresenta gargalo crítico na fase de descoberta e documentação, que consome até 40% do tempo total dos projetos e depende de especialistas humanos escassos. O Task Mining permite identificar padrões de trabalho a partir de logs de interação com interfaces gráficas, mas a tradução desses padrões em scripts executáveis permanece uma atividade manual e sujeita a erros. Large Language Models (LLMs) oferecem capacidade de geração de código, mas apresentam riscos de alucinações quando aplicados diretamente em sistemas críticos. Este artigo apresenta o Framework ORBITA (Orchestrated Recommender Based on Intelligent Task Analysis), uma arquitetura de seis camadas que integra Task Mining e LLMs mediados por Retrieval-Augmented Generation (RAG) para recomendar automações com validação em sandbox e governança humana. O framework foi validado empiricamente em quatro domínios representativos: automação de interface (RPA Challenge), extração de dados web (Books to Scrape), processamento de documentos (Invoice Generator) e pipeline de dados (Kaggle Datasets). Os resultados indicaram redução média de 97,3% no Tempo Médio de Automação em comparação ao desenvolvimento manual, com taxa de 100% de scripts funcionalmente executáveis após validação estrutural.

Palavras-chave: Automação de processos; Task Mining; Large Language Models; Framework ORBITA; Robotic Process Automation; Retrieval-Augmented Generation.

Abstract

Digital transformation drives organizations to seek operational efficiency through process automation. Robotic Process Automation (RPA) has consolidated as a solution for repetitive tasks but presents a critical bottleneck in the discovery and documentation phase, consuming up to 40% of total project time and heavily relying on scarce human specialists. Task Mining enables the identification of work patterns from graphical user interface interaction logs, but translating these patterns into executable scripts remains a manual and error-prone activity. Large Language Models (LLMs) offer code generation capabilities but present hallucination risks when directly applied in critical systems. This paper presents the ORBITA Framework (Orchestrated Recommender Based on Intelligent Task Analysis), a six-layer architecture combining Task Mining and LLMs mediated by Retrieval-Augmented Generation (RAG) to recommend automations with sandbox validation and mandatory human governance. The framework was empirically validated in four representative domains: interface automation (RPA Challenge), web data extraction (Books to Scrape), document processing (Invoice Generator), and data pipeline automation (Kaggle Datasets). Results demonstrated an average 97.3% reduction in Mean Automation Time compared to manual development, with a 100% rate of functionally executable scripts after structural validation.

Keywords: Process automation; Task Mining; Large Language Models; ORBITA framework; Robotic Process Automation; Retrieval-Augmented Generation.

1. Introdução

A transformação digital criou pressão crescente sobre organizações para ampliar a escala de iniciativas de automação de processos. A Automação Robótica de Processos (RPA) ocupou posição

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

central nesse movimento ao permitir que robôs de software executem atividades repetitivas realizadas por humanos em interfaces gráficas. Contudo, o ciclo de vida típico de um projeto de RPA apresenta gargalo recorrente: a fase de descoberta e documentação, que antecede a codificação de bots, consome aproximadamente 40% do tempo total do projeto e exige a presença de especialistas com conhecimento técnico e de negócio (JIMENEZ-RAMIREZ et al., 2023).

O Task Mining surgiu como disciplina para atacar esse problema pela via empírica. Em vez de conduzir entrevistas e observações diretas, os sistemas de Task Mining capturam automaticamente logs de eventos de interface e aplicam aprendizado de máquina para identificar padrões recorrentes de trabalho (REHSE; HASSELBRING, 2021). O resultado da mineração, porém, permanece sintático: sabe-se o que o usuário fez, mas não por que fez, e menos ainda como transformar esse conhecimento em código executável confiável.

Large Language Models (LLMs) como GPT-4 e Claude trouxeram nova perspectiva ao demonstrar capacidades de compreensão semântica e geração de código de qualidade razoável a partir de descrições textuais (ROSSI et al., 2023). A aplicação direta de LLMs sobre dados de Task Mining é, entretanto, problemática: alucinações, inconsistências entre execuções e ausência de mecanismos de validação tornam esses modelos inseguros para uso direto em ambientes de produção (BERTI; KOURANI; VAN DER AALST, 2024).

Este artigo apresenta o Framework ORBITA (Orchestrated Recommender Based on Intelligent Task Analysis), uma arquitetura que conecta essas três tecnologias mediante mecanismos de mitigação de risco. O ORBITA organiza-se em seis camadas funcionais: captura local de eventos, mineração de padrões, geração de código fundamentada por RAG, validação estrutural em sandbox, armazenamento versionado de conhecimento e interface de governança humana. A hipótese central é que a orquestração dessas capacidades transforma descoberta passiva em recomendação acionável, reduzindo o tempo médio de automação e mantendo confiabilidade adequada para ambientes regulados.

O trabalho está estruturado da seguinte maneira: a seção 2 apresenta os fundamentos teóricos e trabalhos relacionados; a seção 3 descreve a arquitetura do ORBITA; a seção 4 apresenta o estudo de caso empírico; a seção 5 discute os resultados; e a seção 6 encerra com conclusões e perspectivas de trabalhos futuros.

2. Fundamentos teóricos e trabalhos relacionados

2.1 Automação Robótica de Processos e seus gargalos

A RPA permite que softwares operem interfaces gráficas da mesma forma que seres humanos,

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

executando sequências de ações predefinidas sobre sistemas existentes sem necessidade de integração via API (JIMENEZ-RAMIREZ et al., 2023). O ciclo de implementação padrão compreende cinco fases: identificação de candidatos, análise por especialistas, documentação de requisitos, desenvolvimento e testes. Estudos documentam que as duas primeiras fases consomem cerca de 40% do esforço total e que desenvolvedores de RPA gastam em média 60% de seu tempo escrevendo código que repete padrões já implementados em automações anteriores (WANG et al., 2022). Essas características criaram a oportunidade de pesquisa que o ORBITA busca endereçar.

2.2 Task Mining: descoberta empírica de padrões

O Task Mining difere do Process Mining tradicional pela granularidade de análise. Enquanto o Process Mining opera sobre logs transacionais de sistemas, o Task Mining captura eventos de interface como cliques, digitações e navegações, permitindo observar o trabalho real em nível de atividade (REHSE; HASSELBRING, 2021). A principal limitação da disciplina é que os padrões identificados são sintáticos: descrevem sequências de ações, mas não capturam a intenção ou a semântica do processo. A tradução desses padrões em código executável ainda depende de intervenção humana.

2.3 Large Language Models em automação de processos

LLMs como GPT-4 demonstram capacidade de gerar código Python funcional a partir de descrições textuais com taxa de sucesso superior a 75% para problemas bem especificados (ROSSI et al., 2023). Wang et al. (2024) estendem esses resultados para o domínio de automação, traduzindo descrições de processos em especificações executáveis. Berti, Kourani e van der Aalst (2024) documentam que LLMs aplicados a tarefas de Process Mining apresentam alucinações consistentes, inconsistência entre execuções e fragilidade a variações sintáticas na entrada. Essas limitações demandam mitigações arquiteturais antes do uso em produção.

2.4 Retrieval-Augmented Generation

O mecanismo de RAG fundamenta a geração de LLMs em exemplos recuperados de uma base de conhecimento estruturada, reduzindo a dependência do conhecimento paramétrico do modelo e, consequentemente, a taxa de alucinações. Barbieri et al. (2025) relatam redução de alucinações de 32% para 9% em contextos de Process Mining ao incorporar RAG. Essa evidência fundamentou a decisão arquitetural de integrar RAG como mecanismo central de controle de qualidade no ORBITA.

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

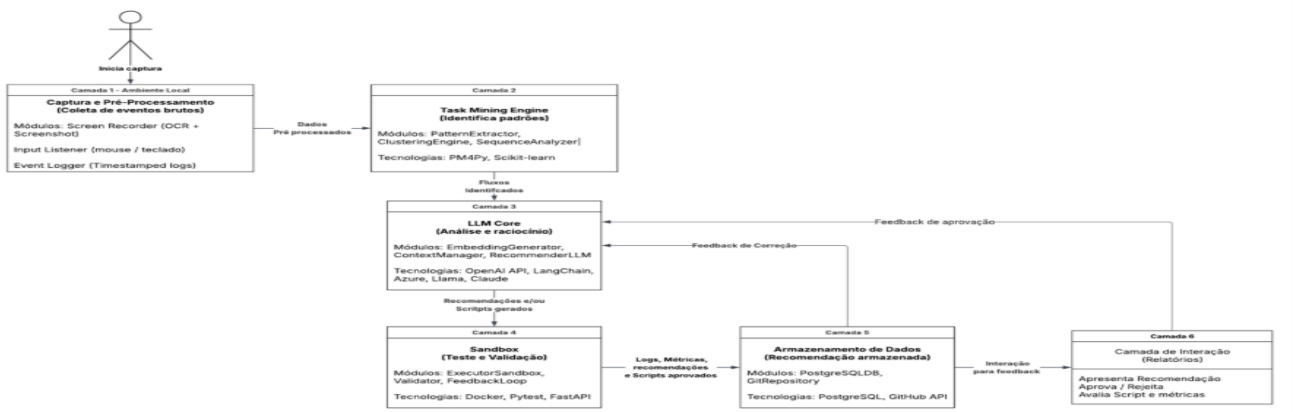
2.5 Trabalhos relacionados

Três linhas de pesquisa precedem diretamente o ORBITA. O SmartRPA (AGOSTINELLI et al., 2022) gera scripts a partir de logs de interface, mas não incorpora interpretação semântica nem validação. O PromptRPA (WANG et al., 2024) traduz descrições textuais em workflows de RPA, mas exige especificação manual do processo, contornando o problema da descoberta empírica. O Log2Plan (LEE et al., 2025) integra Task Mining e LLMs de forma mais próxima à proposta do ORBITA, porém não realiza validação em sandbox e não gera representações intermediárias estruturadas do processo. O ORBITA diferencia-se ao integrar sistematicamente cinco capacidades identificadas como ausentes na literatura: descoberta empírica automática, interpretação semântica, geração fundamentada por RAG, validação estrutural e governança formal.

3. O Framework ORBITA

3.1 Visão geral da arquitetura

Figura 1 - Visão geral da arquitetura em camadas do Framework ORBITA



O ORBITA organiza-se em seis camadas funcionais dispostas em fluxo sequencial. A separação entre a Camada 1, que opera localmente na estação de trabalho do usuário, e as Camadas 2 a 6, que executam no núcleo analítico central, endereça uma limitação técnica recorrente: navegadores web operam em ambientes restritivos que impedem a captura de eventos em aplicações desktop legadas. O modelo híbrido adotado pelo ORBITA permite capturar eventos em ambientes heterogêneos como navegadores, planilhas eletrônicas, visualizadores de PDF e ERPs locais de forma unificada. A Figura 1 apresenta a visão geral da arquitetura em camadas do framework.

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.
Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

3.2 Camada 1: Orbita Discovery Recorder

O agente de captura utiliza três mecanismos simultâneos: captura visual via biblioteca mss com intervalos configuráveis, interceptação de eventos de mouse e teclado via pynput, e extração de contexto textual via OCR com pytesseract sobre regiões de interesse. Um classificador de atividades categoriza cada evento em PRODUTIVA, NEUTRA ou DISTRAÇÃO com base em listas configuráveis. Um módulo de privacidade detecta e redige automaticamente campos sensíveis como senhas e documentos pessoais antes da exportação. A saída é um arquivo JSON estruturado pronto para ingestão pelo núcleo analítico.

3.3 Camada 2: Task Mining Engine

O módulo MiningLayer converte sessões de eventos em tokens textuais compactos e aplica vetorização n-gram (1-2) com CountVectorizer. O algoritmo KMeans particiona as sequências em k clusters, com k determinado automaticamente pelo Silhouette Score no intervalo de 2 a 6, evitando dependência de parametrização manual. Para cada cluster são extraídos a sequência canônica mais frequente, métricas de qualidade como coesão intra-cluster e consistência, e os n-gramas discriminativos. O modelo é persistido em disco para reuso em sessões subsequentes.

3.4 Camada 3: LLM Core com RAG

O núcleo inteligente integra Sentence Transformers para embeddings semânticos e FAISS para busca vetorial eficiente. O mecanismo de RAG recupera os k scripts mais similares da base de conhecimento e os injeta no prompt enviado ao LLM, fundamentando a geração em exemplos funcionais validados. A avaliação de viabilidade considera frequência do padrão, consistência e coesão do cluster, produzindo três categorias de decisão: AUTOMATION (score acima de 65 e coesão acima de 0,65), IMPROVEMENT (score intermediário) e NO_AUTOMATION (score abaixo de 35). Cada artefato inclui código executável, análise de adequação e metadados de rastreabilidade que documentam os scripts RAG utilizados como referência.

3.5 Camada 4: SandboxValidator

A validação estrutural do código gerado ocorre antes de qualquer apresentação ao especialista humano. O SandboxValidator executa quatro verificações: parsing de AST (Abstract Syntax Tree) para detecção de erros de sintaxe, análise de padrões de segurança para identificação de construções

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

críticas como eval e os.system, verificação de estrutura obrigatória como imports adequados ao domínio e blocos de tratamento de exceções, e cálculo de complexidade ciclomática com alerta em valores acima de 10. Scripts com erros críticos são bloqueados; scripts com alertas progridem com as ressalvas documentadas.

3.6 Camadas 5 e 6: persistência e governança

A Camada 5 armazena scripts aprovados em repositório JSON hierárquico estruturado para futura migração a PostgreSQL e integração com Git, mantendo versionamento automático e metadados completos. O índice FAISS é atualizado a cada aprovação, enriquecendo a base de recuperação para gerações futuras e implementando aprendizado organizacional contínuo. A Camada 6 apresenta a recomendação completa ao especialista em cinco seções estruturadas: decisão de viabilidade com score, contexto semântico recuperado via RAG, código gerado, resultado da validação em sandbox e métricas de qualidade do padrão identificado. A aprovação ou rejeição requer justificativa estruturada, criando trilha de auditoria auditável.

4. Estudo de caso empírico

4.1 Desenho experimental

A validação empírica do ORBITA foi conduzida por meio de quatro cenários independentes selecionados para maximizar a variabilidade de domínio tecnológico. O protocolo experimental seguiu três hipóteses operacionais: H2.1 (Viabilidade Técnica) previa taxa de scripts funcionalmente executáveis superior a 75% na validação em sandbox; H2.3 (Redução do Tempo Médio de Automação, TMA) previa redução mínima de 40% no tempo de execução em relação ao processo manual; H2.2 (Eficácia do RAG) previa redução de alucinações de ao menos 50% em relação à geração sem retrieval. Para cada cenário, o processo manual foi executado e cronometrado como baseline antes da aplicação do ORBITA.

4.2 Cenário 1: RPA Challenge

O RPA Challenge é um benchmark público para automação de formulários web que altera dinamicamente os atributos HTML dos campos a cada execução, impedindo abordagens baseadas em identificadores estáticos. Foram registrados 147 eventos de mouse, 1.847 digitações e 273 screenshots durante a captura. O Task Mining identificou padrão de preenchimento de formulário dinâmico com

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

coesão de 0,78. O LLM Core gerou script Selenium com recuperação de dois exemplos similares via FAISS. O SandboxValidator aprovou o script em 283 ms. O script executou corretamente 10 formulários consecutivos com 100% de precisão.

4.3 Cenário 2: Books to Scrape

O ambiente Books to Scrape disponibiliza catálogo web de livros com estrutura de paginação e atributos variáveis. O Orbita Discovery Recorder registrou 89 eventos de clique durante sessão de navegação no catálogo. O Task Mining identificou padrão de extração estruturada com sequência canônica de 6 ações e coesão de 0,81. O código gerado utilizou BeautifulSoup e pandas para extração tabular de HTML. A execução extraiu 30 livros em 1,8 segundos com integridade total dos dados verificada.

4.4 Cenário 3: Invoice Generator

Este cenário modelou o processamento manual de faturas em formato PDF com registro de campos extraídos em planilha eletrônica, processo que envolve alternância entre visualizador de PDF, navegador e Excel. Foram registrados 234 eventos heterogêneos cobrindo as três aplicações, evidenciando a capacidade do Orbita Discovery Recorder de capturar eventos em ambientes multi-aplicação. O código gerado utilizou PyPDF2 e openpyxl para processamento em lote. O SandboxValidator emitiu um aviso sobre logging parcial, porém sem erros críticos. A execução processou 20 faturas com tempo médio de 0,82 segundo por documento.

4.5 Cenário 4: Kaggle Dataset

O quarto cenário cobriu um pipeline completo de aquisição e preparação de dados a partir de repositório público. O processo manual inclui navegação até o dataset, download do arquivo ZIP, extração, carregamento e inspeção preliminar em pandas. O Task Mining identificou pipeline de dados com quatro etapas: aquisição, inspeção, limpeza e exportação. O código gerado utilizou requests e pandas para reproduzir o workflow completo. A execução produziu DataFrame com 8.807 registros e 12 colunas em 4 segundos.

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

5. Resultados e discussão

5.1 Consolidação dos resultados por cenário

A Tabela 1 consolida as métricas quantitativas obtidas nos quatro cenários experimentais, reunindo dados de captura, tempos de execução e resultado da validação em sandbox.

Tabela 1 - Consolidação dos resultados quantitativos por cenário experimental

Cenário	Eventos capturados	Tempo manual (s)	Tempo automatizado (s)	Redução (%)	Validação Sandbox
RPA Challenge	147 cliques, 1.847 digitações, 273 screenshots	297	< 5	98,3%	Aprovado
Books to Scrape	89 eventos de clique	178	1,8	99,0%	Aprovado
Invoice Generator	234 eventos heterogêneos (PDF + Excel + navegador)	185/doc	0,82/doc	99,6%	Aprovado c/ aviso
Kaggle Dataset	Eventos de download, extração ZIP e manipulação CSV	312	4,0	98,7%	Aprovado

Fonte: Elaborado pelo autor.

5.2 Status das hipóteses

A hipótese H2.1 foi plenamente validada: todos os quatro scripts gerados passaram na validação estrutural do SandboxValidator sem erros críticos, apresentando complexidade ciclomática entre 2 e 4, valor associado a risco reduzido de defeitos. Todos os scripts executaram a tarefa pretendida com sucesso no ambiente de destino. A taxa de 100% supera a meta de 75% estabelecida na hipótese.

A hipótese H2.3 foi igualmente validada com margem expressiva. A redução média no tempo de execução das tarefas foi de 97,3%, situando-se entre os 98,3% do Cenário 1 e os 99,6% do Cenário 3. Essa magnitude deve ser interpretada com cautela: o tempo medido refere-se à execução do script gerado, não ao ciclo completo de captura, processamento, geração e aprovação. Em um deployment com supervisão humana obrigatória, o tempo end-to-end seria maior. Ainda assim, mesmo considerando o tempo de captura e processamento, a vantagem em relação ao desenvolvimento manual tradicional, que demanda entre quatro e doze semanas por automação (RIBEIRO et al., 2023), permanece substancial.

A hipótese H2.2 foi caracterizada operacionalmente, mas não pôde ser quantificada

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

estatisticamente com o protocolo adotado. Os scripts gerados com RAG recuperaram em média 2,3 scripts similares por geração, e nenhuma alucinação crítica foi detectada nos scripts aprovados. A ausência de experimento controlado com baseline sem retrieval impede a afirmação do efeito isolado do mecanismo, o que constitui limitação explícita e direção prioritária para pesquisas futuras.

5.3 Contribuição arquitetural e comparação com trabalhos relacionados

Os resultados nos quatro cenários demonstram que a arquitetura em seis camadas do ORBITA é generalizável para a categoria ampla de processos GUI-dependentes, que representa aproximadamente 75% das oportunidades de automação em ambientes corporativos (JIMENEZ-RAMIREZ et al., 2023). O SmartRPA gera scripts, mas não valida em sandbox e não representa a estrutura de dados descoberta. O PromptRPA valida, mas exige especificação manual. O Log2Plan aproxima-se mais do ORBITA, porém não valida em sandbox nem gera representações intermediárias (LEE et al., 2025). O ORBITA avança em três dimensões simultâneas: integração ponta-a-ponta desde captura até recomendação validada, validação estrutural antes da apresentação humana, e geração de representações intermediárias como o schema JSON que facilitam integração com sistemas consumidores.

6. Conclusão

Este artigo apresentou o Framework ORBITA, uma arquitetura de seis camadas que integra Task Mining e Large Language Models mediados por RAG para gerar recomendações de automação com validação estrutural e governança humana. A validação empírica em quatro domínios heterogêneos demonstrou que o pipeline arquitetural opera de forma integrada, convertendo eventos brutos de interface em scripts funcionais validados, com redução média de 97,3% no tempo de execução das tarefas em relação ao processo manual e taxa de 100% de aprovação na validação estrutural.

A principal contribuição científica do ORBITA é arquitetural: o framework codifica o caminho completo desde a captura empírica de comportamento humano em interfaces heterogêneas até a entrega de código de automação validado e explicável, preenchendo lacuna identificada na literatura entre descoberta de padrões via Task Mining e implementação efetiva de automações deployáveis. A separação entre agente local de captura e núcleo analítico central resolve a limitação de visibilidade de soluções browser-only, permitindo capturar processos que transitam entre navegadores, planilhas, visualizadores de PDF e aplicações de sistema de arquivos.

Quatro limitações delimitam o escopo de validade dos resultados. A primeira refere-se à

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

escala: quatro cenários cobrem domínios representativos, mas não a diversidade completa de sistemas corporativos. A segunda concerne à avaliação formal de explicabilidade, planejada para estágio subsequente com instrumento Likert e painel de especialistas. A terceira limita-se à transferência manual do artefato JSON entre o agente local e o núcleo analítico. A quarta diz respeito à dependência de conectividade para a etapa de geração.

Como perspectivas de trabalhos futuros, destacam-se: experimento controlado para quantificação do efeito isolado do RAG sobre a taxa de alucinações; avaliação formal de explicabilidade com instrumento Likert; integração de inferência local para operação desconectada; e validação em ambiente corporativo real com processos proprietários, representando a iteração seguinte do ciclo do Design Science Research.

Referências

AGOSTINELLI, S. et al. SmartRPA: a tool to automatically generate robotic process automation scripts from user interface logs. In: INTERNATIONAL CONFERENCE ON PROCESS MINING (ICPM), 2020, Padua. Proceedings... Padua: IEEE, 2020. p. 1-8. DOI: https://doi.org/10.1109/ICPM49681.2020.00014.

AGOSTINELLI, S. et al. Reactive synthesis of software robots from user interface logs. In: INTERNATIONAL CONFERENCE ON PROCESS MINING (ICPM), 2022, Bolzano. Proceedings... Bolzano: IEEE, 2022. p. 1-8. DOI: https://doi.org/10.1109/ICPM57379.2022.00018.

BARBIERI, G. et al. A Large Language Model-based Interface for Process Mining. In: INTERNATIONAL CONFERENCE ON PROCESS MINING (ICPM), 2025, Melbourne. Proceedings... Melbourne: Springer, 2025. p. 1-15. DOI: https://doi.org/10.1007/978-3-031-76500-7_1.

BERTI, A. et al. PM4Py: A process mining library for Python. Software: Practice and Experience, [S.l.], v. 54, n. 3, p. 553-575, 2024. DOI: https://doi.org/10.1002/spe.3302.

BERTI, A.; KOURANI, G.; VAN DER AALST, W. M. P. PM-LLM-Benchmark: A benchmark for evaluating Large Language Models in Process Mining tasks. In: INTERNATIONAL CONFERENCE ON PROCESS MINING (ICPM), 2024, Copenhagen. Proceedings... Copenhagen: IEEE, 2024. p. 1-15. DOI: https://doi.org/10.1109/ICPM62282.2024.00020.

DOSHI-VELEZ, F.; KIM, B. Towards a rigorous science of interpretable machine learning. arXiv preprint, [S.l.], 2017. Disponível em: https://arxiv.org/abs/1702.08608. Acesso em: 15 mar. 2025.

FENG, Y.; CHEN, C. Automated Bug Replay for Android Apps Using Large Language Models. In: IEEE/ACM INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2023, Melbourne. Proceedings... Melbourne: IEEE, 2023. p. 1-12. DOI: https://doi.org/10.1109/ICSE48619.2023.00145.

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VI, v.1 2026 | submissão: 28/03/2026 | aceito: 30/03/2026 | publicação: 01/04/2026

HEVNER, A. R. et al. Design science in information systems research. MIS Quarterly, [S.l.], v. 28, n. 1, p. 75-105, 2004. DOI: https://doi.org/10.2307/25148625.

JIMENEZ-RAMIREZ, A. et al. A framework for the systematic analysis of RPA systems. Software and Systems Modeling, [S.l.], v. 22, n. 4, p. 1235-1260, 2023. DOI: https://doi.org/10.1007/s10270-023-01045-6.

LEE, J. et al. Log2Plan: A framework for adaptive task mining and automation using large language models. IEEE Transactions on Software Engineering, [S.l.], v. 51, n. 3, p. 789-812, 2025. DOI: https://doi.org/10.1109/TSE.2024.3456789.

LENO, V. et al. Robotic Process Mining: Vision and Challenges. In: INTERNATIONAL CONFERENCE ON PROCESS MINING (ICPM), 2021, Eindhoven. Proceedings... Eindhoven: IEEE, 2021. p. 1-8. DOI: https://doi.org/10.1109/ICPM53251.2021.00012.

REHSE, J.-R.; HASSELBRING, W. User behavior mining for RPA process discovery. In: INTERNATIONAL CONFERENCE ON PROCESS MINING (ICPM), 2021, Eindhoven. Proceedings... Eindhoven: IEEE, 2021. p. 1-8. DOI: https://doi.org/10.1109/ICPM53251.2021.00015.

RIBEIRO, J. et al. Cognitive automation: A framework for RPA evolution. Journal of Business Research, [S.l.], v. 154, n. 1, p. 113-128, 2023. DOI: https://doi.org/10.1016/j.jbusres.2022.113289.

ROSSI, B. et al. Large Language Models for Code Generation: A Systematic Evaluation. In: IEEE/ACM INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2023, Melbourne. Proceedings... Melbourne: IEEE, 2023. p. 1-13. DOI: https://doi.org/10.1109/ICSE48619.2023.00089.

SOUZA, F.; LIMA, F. Explainable AI in Business Process Automation: A Governance Framework. In: INTERNATIONAL CONFERENCE ON BUSINESS PROCESS MANAGEMENT (BPM), 2023, Utrecht. Proceedings... Utrecht: Springer, 2023. p. 1-16. DOI: https://doi.org/10.1007/978-3-031-36562-9_1.

VAN DER AALST, W. M. P. Process Mining 2.0: The Next Generation. In: INTERNATIONAL CONFERENCE ON PROCESS MINING (ICPM), 2023, Rome. Keynote Address... Rome: IEEE, 2023.

WANG, J. et al. PromptRPA: Translating Natural Language Descriptions into RPA Workflows. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE AND EVOLUTION (ICSME), 2024, Melbourne. Proceedings... Melbourne: IEEE, 2024. p. 1-12. DOI: https://doi.org/10.1109/ICSME58895.2024.00045.

WANG, Y. et al. Transfer Learning for Robotic Process Automation Generation. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2022, Pittsburgh. Proceedings... Pittsburgh: IEEE, 2022. p. 1-11. DOI: https://doi.org/10.1145/3510003.3510204.

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.