RPAs e Data Lakes para a Indústria 4.0: Um Estudo de Caso de Ecossistema de Dados Integrados

Arthur Lucas dos S. Bezerra[1], Iranildo S. Batalha[1], Luís Ricardo A. Filho[1],

Clarice M. Almeida[1], Matheus Inácio N. Dantas[1], Nelson A. Gouvêa[1]

[1]AIX Department - LG Electronics do Brasil

{arthur.bezerra, iranildo.batalha, luis.filho,
clarice.almeida, matheus.dantas, nelson.gouvea}@lge.com

Abstract. *The use of RPAs has accelerated process automation in corporate environments but presents performance limitations when dealing with large volumes of data. This context reveals opportunities for improving the scalability and efficiency of such solutions. This paper proposes a distributed, modular, and loosely coupled Data Lake architecture for the collection, storage, and processing of heterogeneous legacy data. The solution leverages open-source tools such as Hadoop, Spark, and Airflow, organized into functional layers. A case study was implemented using production line data at a multinational electronics company, demonstrating the feasibility and benefits of the proposed approach.*

Resumo. *O uso de RPAs tem acelerado a automação de processos em ambientes corporativos, mas apresenta limitações de desempenho diante de grandes volumes de dados. Esse contexto revela oportunidades de melhoria na escalabilidade e eficiência das soluções. Este artigo propõe uma arquitetura de Data Lake distribuída, modular e de baixo acoplamento para coleta, armazenamento e processamento de dados legados heterogêneos. A solução utiliza ferramentas de código aberto como Hadoop, Spark e Airflow, organizadas em camadas funcionais. Foi implementado um estudo de caso com dados de linhas de produção em uma multinacional do segmento de eletrônicos, demonstrando a viabilidade e os benefícios da abordagem proposta.*

1. Introdução

Segundo [Imperva 2024], 49,6% do tráfego de internet é gerado por robôs digitais. Esse cenário reflete uma mudança significativa no perfil de geração de dados, impulsionada por tecnologias de automatização como *Robotic Process Automation* (RPA)[1] e por dispositivos inteligentes conectados. Tais tendências estão alinhadas aos princípios da Indústria 4.0, que valorizam automação, conectividade e análise de dados em tempo real [Pereira and Simonetto 2018].

Esse novo panorama resultou em um ecossistema de dados cada vez mais amplo, veloz e variado, caracterizado pelos 5Vs do *Big Data* (volume, variedade, velocidade, veracidade e valor), o que intensifica os desafios relacionados à ingestão, organização, persistência, interoperabilidade, escalabilidade e governança [Khine, Pwint Phyu and Wang, Zhao Shun 2018].

[1]Automação de tarefas de serviço que replicam o trabalho realizado por humanos [Ribeiro et al. 2021]

No contexto industrial, por exemplo, a crescente adoção de RPA para automatizar tarefas e integrar sistemas legados contribui significativamente para essa dinâmica, gerando fluxos contínuos de dados heterogêneos. Tais dados, embora essenciais para a otimização de processos, monitoramento e decisões ágeis, podem apresentar inconsistências e formatos diversos devido à sua coleta automatizada, exigindo governança rigorosa desde o ponto de origem [Ribeiro et al. 2021].

Nesse contexto, o trabalho de [Vasconcelos and Coutinho 2024] demonstra que soluções tradicionais baseadas em bancos de dados relacionais, como os *Data Warehouses* (DW), não atendem adequadamente às demandas de dados heterogêneos e em constante transformação. Como alternativa, destacam-se os *Data Lakes* (DL), definidos por [Nargesian et al. 2019] como repositórios para grandes volumes de dados brutos, estruturados ou não, com processamento e análise flexíveis, oferecendo armazenamento multiformato e arquiteturas de organização de dados. No entanto, essa abordagem exige uma governança eficaz baseada em metadados. A ausência desse controle acarreta a formação de um *data swamp*, isto é, um repositório desorganizado e ineficiente [Khine, Pwint Phyu and Wang, Zhao Shun 2018]. Apesar da consolidação dos conceitos fundamentais, observa-se uma lacuna na literatura quanto à aplicação prática de arquiteturas eficazes em ambientes automatizados de produção de dados.

Portanto, este trabalho propõe uma arquitetura de DL orientada a zonas, projetada para ambientes automatizados de geração de dados por meio de RPAs. A solução prioriza a padronização, rastreabilidade e controle de qualidade desde a ingestão, promovendo modularidade e separação de responsabilidades no pipeline de dados. A principal contribuição deste trabalho consiste na concepção e validação prática de uma abordagem arquitetural alinhada aos princípios da Indústria 4.0, que assegura a preservação de dados brutos e a governança de dados heterogêneos em cenários de alta automação, oferecendo um modelo replicável voltado à qualidade, integridade e conformidade no ciclo de vida dos dados.

Este trabalho foi organizado como segue: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 descreve a arquitetura proposta; a Seção 4 discute sua implementação em um estudo de caso; e a Seção 5 traz as considerações finais e perspectivas futuras.

2. Trabalhos Relacionados

[Tito et al. 2020] apresenta uma arquitetura de DL voltada para o setor de saúde, com ênfase na acessibilidade a usuários leigos. A solução permite a ingestão e visualização de dados tabulares fornecidos pelos usuários, obtendo boa aceitação conforme o modelo TAM (*Technology Acceptance Model*). Entretanto, o pré-processamento realizado antes da ingestão compromete a preservação dos dados brutos, princípio fundamental dos DLs [Nargesian et al. 2019]. Além disso, a abordagem limita-se a dados tabulares e não especifica uma organização interna, restringindo sua aplicabilidade em cenários heterogêneos e de larga escala.

[Yang et al. 2021] propõe uma arquitetura de DL para dados do setor elétrico, integrando diversas ferramentas do ecossistema Apache (e.g., Hadoop, Spark, Kafka) para lidar com cenários complexos e dados em tempo real. Embora robusta para *streaming* estruturado, sua implementação revela considerável complexidade, em parte devido à diversidade tecnológica envolvida. Observa-se também exploração limitada no tratamento de formatos variados (e.g., imagens, logs) e na definição de governança para tal hetero-

geneidade, aspecto crucial para arquiteturas DL flexíveis frente à diversidade de dados.

No enfoque da automação da ingestão, [Kothandapani 2021] explora RPA para coleta e armazenamento automático de dados heterogêneos oriundos de múltiplas fontes no DL, além do processamento com Apache Spark. Propõe ainda APIs[2] para padronizar a recepção dos dados, melhorando a organização e automação dos pipelines. Contudo, a abordagem concentra todas as etapas (da ingestão ao processamento) no RPA, limitando a separação clara de conceitos e responsabilidades ao longo do pipeline.

Esses trabalhos revelam avanços relevantes em arquiteturas DL, contribuindo para usabilidade em domínios específicos [Tito et al. 2020], gestão de dados em tempo real com ferramentas Apache [Yang et al. 2021] e automação da ingestão heterogênea via RPA [Kothandapani 2021]. Contudo, persistem lacunas importantes: preservação dos dados brutos com tratamento e governança de formatos heterogêneos [Tito et al. 2020, Yang et al. 2021]; equilíbrio entre robustez funcional e simplicidade de implantação [Yang et al. 2021]; e maior modularidade com clara separação de responsabilidades em pipelines automatizados via RPA, assegurando zonas de dados que garantam qualidade, rastreabilidade e governança em ambientes produtivos [Kothandapani 2021].

Para suprir tais lacunas, propõe-se uma arquitetura automatizada de DL que preserva dados brutos e viabiliza tratamento e governança de dados heterogêneos. Tal estrutura utiliza RPAs para ingestão eficiente de múltiplos formatos, enfatizando a separação de responsabilidades, modularidade do pipeline e estrutura organizacional detalhada por zonas de dados. Ao equilibrar robustez e menor complexidade de implantação, a proposta configura uma alternativa viável para ambientes produtivos de geração automatizada, promovendo qualidade, rastreabilidade e conformidade ao longo do ciclo dos dados.

3. Solução Proposta

Na organização analisada, os dados são extraídos de múltiplos sistemas e armazenados em formatos diversos, majoritariamente em planilhas Excel, sem um repositório centralizado. Essa dispersão compromete a rastreabilidade, a persistência e a governança dos dados. Para resolver essas limitações, propõe-se uma arquitetura de Data Lake com foco em centralização, padronização e controle, visando maior eficiência, eliminação de redundâncias e acesso confiável à informação. A Figura 1 apresenta essa arquitetura, dividida em três camadas: Ingestão, Armazenamento e Processamento.

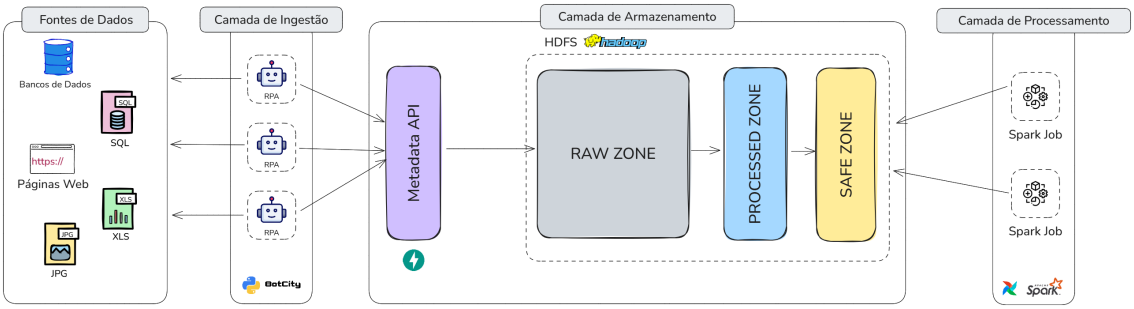

Figure 1. Proposta da arquitetura de DL

[2]*Application Programming Interface*, interfaces programáticas que permitem a interação entre sistemas e componentes de software

Na **Camada de Ingestão de Dados** estão localizadas as automações responsáveis pela coleta de dados a partir das fontes originais, predominantemente sistemas legados da organização, embora não se restrinjam a estes. Tais automações são implementadas em Python aliado ao *framework* BotCity[3] que simplifica o desenvolvimento estruturado dessas soluções. Quando uma automação executa sua tarefa de coleta, realiza a etapa de armazenamento dos dados, integrando-se diretamente com a camada subsequente da arquitetura por meio da API de Metadados.

Camada de Armazenamento de Dados foi estruturada com dois componentes principais: HDFS[4] e a API de Metadados. A implementação da API, que segue o protocolo HTTP no padrão REST, foi uma decisão estratégica para promover a escalabilidade, a simplicidade e a interoperabilidade entre os módulos da arquitetura, conforme discutido por [Kothandapani 2021]. Por meio de seus serviços, gerencia as operações de armazenamento e o registro de metadados essenciais para a governança, assegurando que cada dado seja rastreável desde a ingestão.

A organização dos dados foi estruturada com base em uma versão adaptada da arquitetura de zonas descrita por [Rodrigues and Mello 2022], segmentada da seguinte forma:

I. *Raw Zone* — implementada como a zona de armazenamento de arquivos brutos gerados por automações, incluindo imagens e dados tabulares;
II. *Processed Zone* — configurada como zona intermediária, onde os dados processados inicialmente são versionados e preparados para transformações futuras.
III. *Safe Zone* — estabelecida como zona final, com dados padronizados em formato tabular (Parquet[5], CSV ou SQL) e estruturados conforme os requisitos de consumo.

A movimentação dos dados entre essas zonas foi implementada de forma controlada e sistemática, tendo a *Raw Zone* sido definida como o repositório central e imutável de todos os dados brutos ingeridos. Essa definição permitiu que as demais zonas, como a *Processed Zone* e a *Safe Zone*, fossem derivadas direta e indiretamente da *Raw Zone*, assegurando a reprodutibilidade dos processos de transformação. A adoção desse modelo promoveu maior robustez arquitetural ao implementar o versionamento e a rastreabilidade. Ambas são consideradas práticas de governança de dados consolidadas na literatura [Giebler et al. 2021], garantindo a integridade do ciclo de vida da informação.

Por fim, a **Camada de Processamento de Dados** foi implementada com a responsabilidade de aplicar as transformações necessárias aos dados, em conformidade com as regras de negócio e os requisitos das aplicações consumidoras. Essa etapa abrangeu desde operações de limpeza e padronização até a consolidação de múltiplas fontes e a derivação de estruturas analíticas.

As transformações foram executadas por meio do Apache Spark, ferramenta adotada para possibilitar o processamento distribuído em larga escala dentro do ecossistema Hadoop, garantindo alta performance mesmo diante de grandes volumes de dados, conforme descrito por [Minh et al. 2024]. A orquestração das tarefas foi realizada utilizando o Apache Airflow[6], que assegurou rastreabilidade, modularidade e controle na execução

[3]https://documentation.botcity.dev/frameworks/

[4]*Hadoop Distributed File System*, um sistema de arquivos distribuído do ecossistema Hadoop

[5]Formato de arquivo colunar e aberto, projetado para armazenar e recuperar dados de forma eficiente

[6]Ferramenta de código aberto para gerenciamento de workflow [Shukla 2022]

dos fluxos de trabalho, características também observadas em [Shukla 2022].

O fluxo de processamento tem início com um *HDFS Sensor* no DAG[7] do Airflow, responsável por monitorar a chegada de novos arquivos em um diretório específico da *Raw Zone*, organizados conforme metadados previamente definidos. Ao identificar novos dados, um *pipeline* do Spark é acionado para executar operações de limpeza e transformação, como remoção de duplicatas, padronização de formatos de data e eliminação de valores nulos. Como resultado, são geradas duas saídas no formato Parquet: uma nova versão transacional, que assegura rastreabilidade e reprodutibilidade, e um conjunto agregado de dados históricos, ambos armazenados na *Processed Zone*. Na etapa final, os dados são consolidados na *Safe Zone*, onde passam por transformações adicionais apropriadas para consumo e são disponibilizados às aplicações finais, de acordo com as regras de negócio. A Figura 2 ilustra esse fluxo.

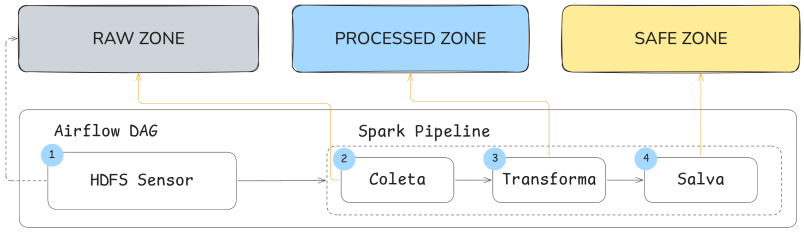

Figure 2. Fluxo de processamento de dados

4. Implementação da proposta a partir de um Estudo de Caso

Esta seção valida experimentalmente a arquitetura de DL proposta, demonstrando sua aplicabilidade em um cenário real de ingestão automatizada de dados por RPA, a eficiência das tecnologias de armazenamento e processamento adotadas e o potencial de escalabilidade observado.

Para validar a arquitetura, integrou-se ao DL um RPA operacional que coleta dados da linha de produção e os envia à *Raw Zone* via API de Metadados. Diariamente, a automação é responsável por coletar cerca de 20 mil registros (podendo atingir picos de 100 mil) distribuídos em 13 colunas. Atualmente, a base conta com aproximadamente 3,6 milhões de registros desde o início da coleta. Considerando a importância desses dados para os indicadores e a quantidade massiva de dados, o fluxo foi representativo. A integração mostrou que a coleta, armazenamento e processamento inicial ocorreram de forma contínua e estável, comprovando a eficácia da arquitetura na gestão e preparação dos dados para análises em ambiente produtivo.

A separação de responsabilidades entre os componentes da arquitetura trouxe ganhos em manutenibilidade e escalabilidade. Antes centralizado nos RPAs, todo o fluxo de coleta, tratamento e persistência foi dividido: os RPAs agora se limitam à extração e envio dos dados brutos, enquanto o pipeline do DL assume o processamento, aplicando operações como remoção de duplicatas e valores nulos.

[7] *Directed Acyclic Graph*, representa uma pipeline de tarefas com dependências definidas e sem ciclos.

5. Resultados

O sistema de versionamento foi exercitado nesse cenário. Cada lote diário coletado pelo RPA foi inicialmente alocado em um subdiretório *temp* na *Raw Zone*, sinalizando pendência de processamento. Após o pipeline ser executado, os dados foram consolidados na partição principal, com seus metadados registrados para garantir rastreabilidade. Na *Processed Zone*, cada execução do pipeline gerou uma nova versão em Parquet, armazenada em um diretório de transações. Esse mecanismo permitiu o rastreamento completo do ciclo de vida dos dados, da coleta à transformação.

Table 1. Comparativo de eficiência de armazenamento por formato de dados.

Formato	Tamanho Ocupado (MB)	Redução com Parquet (%)
PostgreSQL	889	71,88%
CSV	646	62,3%
Parquet	250	-

Os resultados observados na Tabela 1 confirmam a alta eficiência do formato Parquet na otimização do armazenamento, com reduções de espaço superiores a 62% em comparação ao CSV e PostgreSQL.

A arquitetura proposta foi concebida com a escalabilidade horizontal como fundamento (Modelo B na Figura 3), utilizando tecnologias como Apache Spark que distribuem dados e tarefas entre múltiplos nós de um cluster. Em contraste, o Modelo A (escalabilidade vertical) exige máquinas cada vez mais potentes para lidar com maiores volumes de dados, uma característica típica de bancos relacionais. A adoção do Spark no pipeline entre as zonas do DL assegurou a capacidade da arquitetura de se adaptar ao crescimento do volume e da complexidade analítica com maior eficiência, resiliência e melhor custo-benefício.

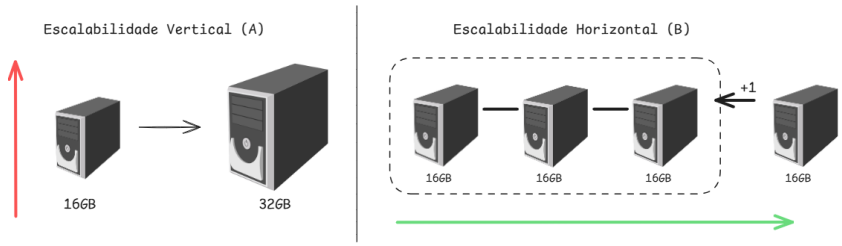

Figure 3. Escalabilidade Vertical vs Horizontal

6. Considerações Finais

Este trabalho apresentou uma arquitetura de Data Lake orientada a zonas para a ingestão automatizada de dados via RPA, focando em governança, modularidade e rastreabilidade. Validada em um estudo de caso na Indústria 4.0, a solução demonstrou robustez, escalabilidade horizontal e eficácia na otimização do armazenamento. Como trabalhos futuros, a pesquisa será aprofundada com análises quantitativas de desempenho, por meio de benchmarks que comparem o acesso a dados brutos versus processados. Adicionalmente, será investigado o custo computacional das rotinas de limpeza para detalhar o impacto do tratamento de dados.

Referências

Giebler, C., Gröger, C., Hoos, E., Eichler, R., Schwarz, H., and Mitschang, B. (2021). The data lake architecture framework: A foundation for building a comprehensive data lake architecture.

Imperva (2024). Imperva 2024 bad bot report. Technical report, Imperva Inc. Accessed: 2025-05-19.

Khine, Pwint Phyu and Wang, Zhao Shun (2018). Data lake: a new ideology in big data era. *ITM Web Conf.*, 17:03025.

Kothandapani, H. P. (2021). Integrating robotic process automation and machine learning in data lakes for automated model deployment, retraining, and data-driven decision making.

Minh, T. P., Quang, H. H., and Manh, T. N. (2024). A zone-based data lake architecture for smart crop farming in vietnam: A strategic perspective. In *Proceedings of the 2nd International Conference - Resilience by Technology and Design (RTD 2024)*, pages 29–44. Atlantis Press.

Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q., and Arocena, P. C. (2019). Data lake management: challenges and opportunities. *Proc. VLDB Endow.*, 12(12):1986–1989.

Pereira, A. and Simonetto, E. (2018). Indústria 4.0: Conceitos e perspectivas para o brasil. *Revista da Universidade Vale do Rio Verde*, 16(1). Doutorando e professor do Programa de Pós-Graduação em Administração, UFSM.

Ribeiro, J., Lima, R., Eckhardt, T., and Paiva, S. (2021). Robotic process automation and artificial intelligence in industry 4.0 – a literature review. *Procedia Computer Science*, 181:51–58. CENTERIS 2020 - International Conference on ENTERprise Information Systems / ProjMAN 2020 - International Conference on Project MANagement / HCist 2020 - International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020.

Rodrigues, J. and Mello, R. (2022). Um estudo sobre arquiteturas e metadados em data lakes. In *Anais da XVII Escola Regional de Banco de Dados*, pages 131–134, Porto Alegre, RS, Brasil. SBC.

Shukla, S. (2022). Developing pragmatic data pipelines using apache airflow on google cloud platform. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*, 10:1–8.

Tito, L., Motinha, C., Santiago, F., Ocaña, K., Bedo, M., and de Oliveira, D. (2020). Xi-dl: um sistema de gerência de data lake para monitoramento de dados da saúde. In *Anais do XXXV Simpósio Brasileiro de Bancos de Dados*, pages 151–156, Porto Alegre, RS, Brasil. SBC.

Vasconcelos, F. F. and Coutinho, F. J. (2024). Data lakehouses para a análise de dados geoespaciais em larga escala. In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 722–728, Porto Alegre, RS, Brasil. SBC.

Yang, C.-T., Chen, T.-Y., Kristiani, E., and Wu, S. F. (2021). The implementation of data storage and analytics platform for big data lake of electricity usage with spark. *The Journal of Supercomputing*, 77(6):5934–5959.