PBPC
ISSN 2674-9432
Qualis A3 CAPES 2021-2024
DOI - Crossref
Latindex
Indexado no Google Acadêmico

Uma avaliação empírica de técnicas de particionamento de dados em pipelines de dados distribuídos

Evandro Costa Ferreira[1], Glauber da Rocha Balthazar[2]

https://doi.org/10.36557/2674-9432.2026v5n3p1218-1233
Artigo recebido em 16 de Março e publicado em 16 de Maio de 2026

ARTIGO ORIGINAL

RESUMO

Este artigo apresenta um estudo sobre a otimização de pipelines de dados em ambientes distribuídos, com foco na eficiência de recursos e redução de custos operacionais. O objetivo foi avaliar o impacto de diferentes estratégias de particionamento em grandes volumes de dados utilizando a plataforma Databricks integrada ao Azure Data Lake Storage Gen2. Foi utilizado um dataset de aproximadamente 120 GiB, analisado em três cenários: (i) sem particionamento, (ii) particionamento tradicional e (iii) particionamento por liquid clustering. As métricas avaliadas incluíram tempo de processamento, uso de CPU, uso de memória e custo financeiro. Os resultados demonstraram que o liquid clustering reduziu o tempo médio de execução de 547,03 s para 206,77 s, representando uma redução superior a 60% em relação ao cenário sem particionamento e de aproximadamente 35% em relação ao particionamento tradicional, além de diminuir o custo operacional em mais de 50%. A análise estatística (ANOVA de Welch) indicou diferenças significativas entre os cenários ($p < 0,001$), corroborando a superioridade da abordagem adaptativa. Conclui-se que a escolha da estratégia de particionamento é determinante para a eficiência de pipelines de dados em ambientes distribuídos.

Palavras-chave: big data; databricks; liquid clustering; particionamento de dados; pyspark. azure data lake store gen2.

ABSTRACT

This article presents a study on the optimization of data pipelines in distributed environments, focusing on resource efficiency and operational cost reduction. The objective was to evaluate the impact of different partitioning strategies on large-scale data processing using the Databricks platform integrated with Azure Data Lake Storage Gen2. A dataset of approximately 120 GiB was used and analyzed under three scenarios: (i) no partitioning, (ii) traditional partitioning, and (iii) partitioning using liquid clustering. The evaluated metrics included processing time, CPU usage, memory usage, and financial cost. The results showed that liquid clustering reduced the average execution time from 547.03 s to 206.77 s, representing a reduction of over 60% compared to the non-partitioned scenario and approximately 35% compared to traditional partitioning, in addition to reducing operational costs by more than 50%. Statistical analysis using Welch's ANOVA indicated significant differences between scenarios ($p < 0.001$), supporting the superiority of the adaptive approach. It is concluded that the choice of partitioning strategy is a determining factor for the efficiency of data pipelines in distributed environments.

Keywords: big data; Databricks; liquid clustering; data partitioning; PySpark; Azure Data Lake Storage Gen2

Instituição afiliada
[1] Federal Institute of São Paulo (IFSP), Brazil. Orcid: 0009-0001-1884-8320. E-mail: evandrocf4@gmail.com.br
[2] Federal Institute of São Paulo (IFSP), Brazil. Orcid: 0000-0002-1993-6621. E-mail: glauber.balthazar@ifsp.edu.br

Autor correspondente: *Glauber da Rocha Balthazar* glauber.balthazar@ifsp.edu.br

1 INTRODUÇÃO

O aumento contínuo na geração e no volume de dados tem exigido soluções capazes de processá-los de forma eficiente e escalável, impulsionando o uso de arquiteturas baseadas em computação distribuída e serviços em nuvem. Nesse contexto, tecnologias como Apache Hadoop, Apache Spark e Kubernetes, associadas a provedores como Amazon Web Services (AWS), Google Cloud e Microsoft Azure, disponibilizam ambientes computacionais flexíveis e elásticos, adequados às demandas de processamento em larga escala. Diante desse cenário, os pipelines de dados tornam-se elementos fundamentais, pois estruturam o fluxo automatizado de coleta, transformação, armazenamento e disponibilização de dados, influenciando diretamente o desempenho, a confiabilidade e os custos das operações analíticas (SHAIKH et al., 2019; AUNG e MAW, 2017).

Dentre as plataformas amplamente utilizadas, destaca-se o Databricks, que integra o Apache Spark em um ambiente unificado voltado à engenharia e ciência de dados, oferecendo recursos nativos que viabilizam a otimização de pipelines. Entre essas técnicas, o particionamento de dados tem sido amplamente explorado como estratégia para melhoria de desempenho. Mais recentemente, abordagens adaptativas, como o liquid clustering, têm sido propostas com o objetivo de organizar dinamicamente os dados em data lakes com base em padrões de acesso e consulta, potencializando a eficiência do processamento.

Embora a literatura (CHERUKURI et al., 2021; DARAM, 2021; SHAIKH et al., 2019) reconheça a importância de técnicas de otimização em ambientes distribuídos, particularmente no que se refere à melhoria de desempenho e à redução de custos em pipelines de dados ainda são limitados os estudos que avaliam empiricamente, em cenários reais de grande escala, o impacto comparativo entre estratégias tradicionais e abordagens adaptativas de particionamento. Nesse sentido, este trabalho parte da hipótese de que a aplicação de técnicas de particionamento — em especial aquelas baseadas em organização dinâmica dos dados — tende a melhorar significativamente o desempenho de consultas analíticas, reduzindo tanto a latência quanto os custos operacionais de armazenamento e processamento.

Dessa forma, o objetivo deste estudo foi avaliar o impacto de diferentes

estratégias de particionamento na eficiência de pipelines de dados em ambientes distribuídos. Foram analisados três cenários distintos: (i) ausência de particionamento, (ii) particionamento tradicional e (iii) particionamento utilizando liquid clustering. A avaliação foi conduzida com base em métricas de desempenho relevantes em contextos de Big Data, incluindo tempo de processamento, utilização de CPU, utilização de memória e custo financeiro. Por fim, a relevância desta pesquisa está associada à crescente necessidade de otimização de recursos em projetos de Big Data, especialmente em ambientes de nuvem caracterizados por grande volume de dados e variabilidade nos padrões de acesso. Nesse contexto, este estudo contribuiu para a literatura ao apresentar uma avaliação empírica comparativa, em ambiente real de grande escala, do impacto de diferentes estratégias de particionamento na eficiência de pipelines de dados distribuídos. Diferentemente de abordagens predominantemente teóricas ou baseadas em cenários simulados, a pesquisa demonstra, por meio de métricas operacionais e validação estatística, a superioridade de técnicas adaptativas, como o liquid clustering, em relação a métodos tradicionais. Dessa forma, os resultados ampliam as discussões existentes na literatura (BHATTACHARYA et al., 2021; YANG et al., 2019) ao evidenciar que a organização dinâmica dos dados, orientada por padrões de acesso, constitui um fator determinante para a otimização de desempenho e redução de custos, oferecendo subsídios tanto para o avanço científico quanto para a tomada de decisão em arquiteturas de pipelines de dados.

2 FUNDAMENTAÇÃO TEÓRICA

O presente estudo está fundamentado nos princípios de processamento distribuído de dados em larga escala, com ênfase nas tecnologias Apache Spark e Databricks, que representam o estado da arte em ambientes modernos de Big Data. O Apache Spark é reconhecido como um motor unificado e de alto desempenho para processamento de dados distribuídos, com suporte a diversas linguagens e APIs otimizadas para análise em tempo real e em lote (ZAHARIA et al., 2016). Sua arquitetura baseada em Resilient Distributed Datasets (RDDs) e DataFrames o torna ideal para tarefas intensivas com big data, como as realizadas neste trabalho.

O Databricks, por sua vez, oferece uma camada adicional de abstração e

automação sobre o Spark promovendo melhorias em escalabilidade, gerenciamento de recursos e usabilidade por meio de notebooks colaborativos e recursos como Delta Tables e liquid clustering (DAMJI et al., 2020). As práticas recomendadas para otimização de cargas de trabalho e arquitetura de pipelines em Databricks, discutidas no guia oficial da plataforma (Databricks, 2024) foram fundamentais para a definição da abordagem experimental adotada.

Diversos estudos correlatos (CHERUKURI et al., 2021; DARAM, 2021) reforçam a importância da otimização de pipelines em ambientes distribuídos para melhorar desempenho e reduzir custos computacionais. Portanto, a aplicação de técnicas adequadas de particionamento, ordenação e alocação de recursos é constantemente apontada como essencial para a eficiência operacional (SHANMUKHA Eeti et al., 2021; PAMADI et al., 2021). Neste sentido, a escolha entre estratégias tradicionais e novas abordagens, como o liquid clustering, torna-se uma variável crítica na arquitetura de dados moderna.

Além disso, aspectos relacionados à automação, qualidade dos dados e consistência de leitura também foram considerados com base nas contribuições de Eeti, Jain e Goel (2020) e Pamadi et al. (2021) reforçando a necessidade de boas práticas em projetos que envolvem grandes volumes de dados em nuvem. As práticas e resultados obtidos nesta pesquisa alinham-se com as tendências apontadas na literatura (CHERUKURI et al., 2021; DARAM, 2021; SHAIKH et al., 2019; AUNG e MAW, 2017), contribuindo para sua validação e replicabilidade.

3 METODOLOGIA DA PESQUISA

O presente trabalho adotou uma abordagem experimental aplicada, com foco na análise comparativa de estratégias de particionamento de dados em pipelines distribuídos, utilizando a plataforma Databricks integrada à Microsoft Azure. As etapas metodológicas foram conduzidas com o objetivo de simular um ambiente real de projetos de Ciência de Dados e Engenharia de Dados baseado em trabalhos correlatos (CHERUKURI et al., 2021; DARAM, 2021). Por exemplo, estudos como o de Alagarsamy (2025) exploram o impacto transformador da inteligência artificial na otimização de consultas e dados em sistemas financeiros, destacando a importância de conjuntos de

consultas representativos para avaliar técnicas de aprendizado de máquina em bancos de dados financeiros de alta frequência.

Inicialmente, foi realizada a seleção do conjunto de dados a partir da base pública do Cadastro Nacional da Pessoa Jurídica (CNPJ), disponibilizada pela Receita Federal do Brasil[1] por meio de três etapas de tratamento de daods: ingestão, limpeza e padronização dos dados. Para tanto, o conjunto, com volume superior a 120 GiB, foi escolhido por representar um cenário realista de Big Data, composto por múltiplos arquivos compactados no formato .zip. Após a extração, os arquivos apresentavam dados em formato textual, com campos delimitados por ponto e vírgula (;) e presença de cabeçalhos em todas as estruturas.

Os dados estavam organizados por domínio temático, conforme o conteúdo das informações, incluindo arquivos específicos para Estabelecimentos, Empresas, CNAEs (Classificação Nacional de Atividades Econômicas), entre outros. Em seguida, os dados foram ingeridos no Azure Data Lake Storage Gen2 (ADLS Gen2) por meio de scripts em Python executados na plataforma Databricks, explorando o paralelismo e a escalabilidade do ambiente distribuído. Para garantir a consistência dos resultados, todas as etapas foram executadas sob a mesma configuração de cluster no Databricks, composta por uma instância de driver e duas instâncias de workers do tipo Standard_DS3_v2, com 14 GiB de memória RAM e 4 vCPUs cada.

Na sequência, foi realizada a etapa de limpeza e modelagem dos dados, incluindo padronização de tipos, tratamento de inconsistências e normalização das informações, preparando o conjunto de dados para a aplicação das diferentes estratégias de particionamento. Com os dados tratados, foram criadas tabelas no formato Delta Lake para três cenários distintos: (i) sem particionamento; (ii) com particionamento tradicional baseado em colunas de alta cardinalidade; e (iii) com particionamento utilizando liquid clustering, técnica dinâmica de organização de dados nativa do Databricks.

Esses três cenários possibilitaram a execução de 20 consultas. A definição desse conjunto foi inspirada em práticas de benchmarks e estudos de caso que utilizam cargas de consulta representativas para avaliar o desempenho de sistemas de dados e

[1] https://arquivos.receitafederal.gov.br/dados/cnpj/dados_abertos_cnpj

algoritmos em cenários realistas. Por exemplo, estudos como o de Alagarsamy (2025) exploram a relevância de conjuntos de consultas na avaliação de desempenho em sistemas financeiros de alta frequência. Dessa forma, a seleção de 20 consultas buscou representar uma variedade de operações complexas, típicas de ambientes reais de ciência de dados e engenharia de dados. As consultas incluíram operações como junções entre múltiplas tabelas, ordenações, agregações, filtros encadeados e cálculos derivados, permitindo uma avaliação abrangente do desempenho dos pipelines sob diferentes estratégias de particionamento. A alocação de recursos computacionais foi cuidadosamente dimensionada, considerando tipos de clusters, número de nós e políticas de escalabilidade, com base em testes preliminares, a fim de garantir equidade na comparação entre os cenários.

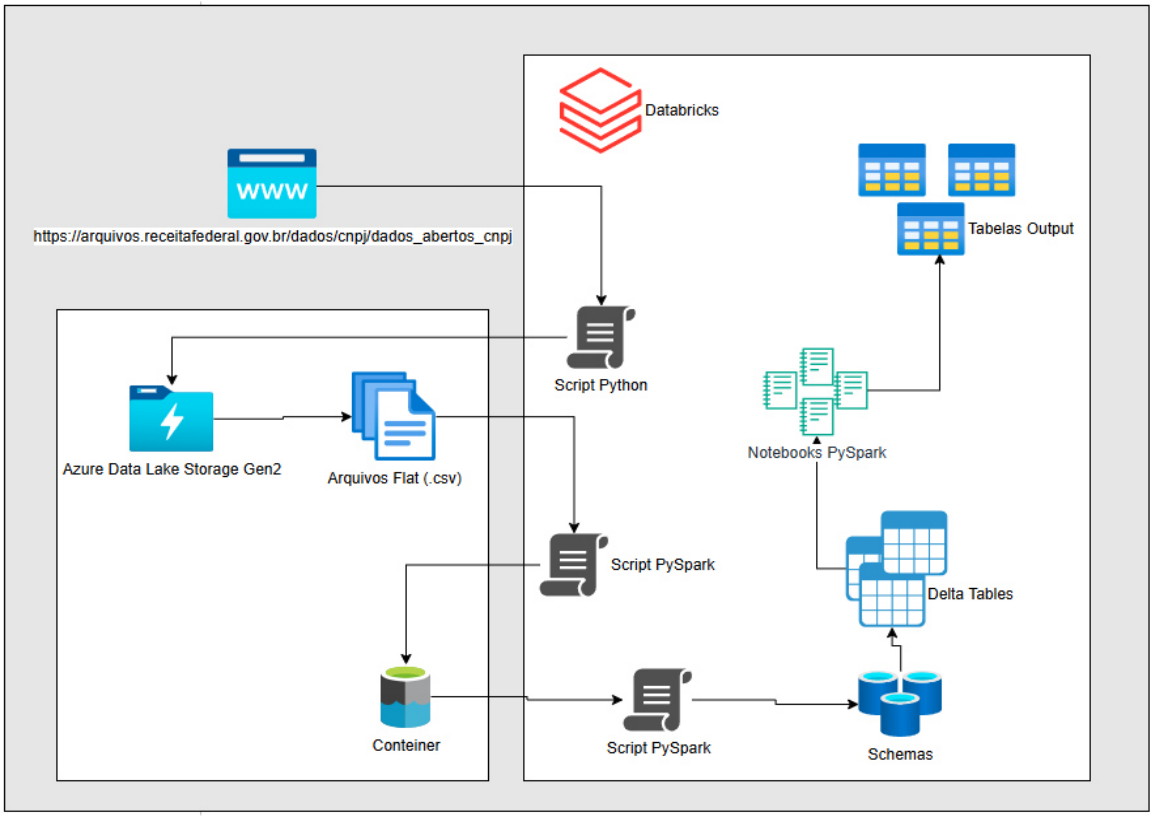
Por fim, foram coletadas métricas de desempenho para cada cenário, incluindo tempo de processamento, utilização de CPU, utilização de memória e custo financeiro. Essas métricas foram obtidas por meio dos logs e das ferramentas nativas de monitoramento do Databricks. As análises foram conduzidas de forma quantitativa, com múltiplas execuções por cenário, assegurando a confiabilidade estatística dos resultados. A avaliação de desempenho foi realizada por meio de estatística descritiva (medidas de tendência central) (SILVA et al., 2025) e inferencial utilizando a ANOVA de Welch (MOORE, 2023).

4 RESULTADOS E DISCUSSÃO

Após as etapas de ingestão, limpeza e padronização dos dados da base do CNPJ, o conjunto final apresentou aproximadamente 120 GiB de dados estruturados no formato parquet. Em seguida, foram criadas tabelas no formato Delta Lake para os três cenários de análise — sem particionamento, particionamento tradicional e particionamento via liquid clustering — organizadas em um único catálogo com três schemas, garantindo a consistência estrutural entre os experimentos. As mesmas transformações foram aplicadas em todos os cenários, incluindo normalização de tipos e consolidação de registros, assegurando a comparabilidade dos resultados. Além disso as tabelas foram construídas respeitando a natureza dos dados, por exemplo, uma tabela para Classificação Nacional das Atividades Econômicas (CNAEs), outra para empresas, estabelecimentos, etc.

A infraestrutura utilizada foi configurada no ambiente Databricks com clusters dimensionados de acordo com o volume e a complexidade das consultas. Para garantir a equidade experimental, todos os cenários foram executados sob configurações idênticas de cluster (instâncias com 32 vCPUs e 256 GB de RAM, com auto escalonamento limitado a no máximo 10 nós), eliminando vieses decorrentes de diferenças na alocação de recursos computacionais. A Figura 1 ilustra a arquitetura de execução, a preparação para os três cenários e o output final das consultas no formato de tabelas delta.

Figura 1 - Arquitetura do trabalho



Fonte: os autores.

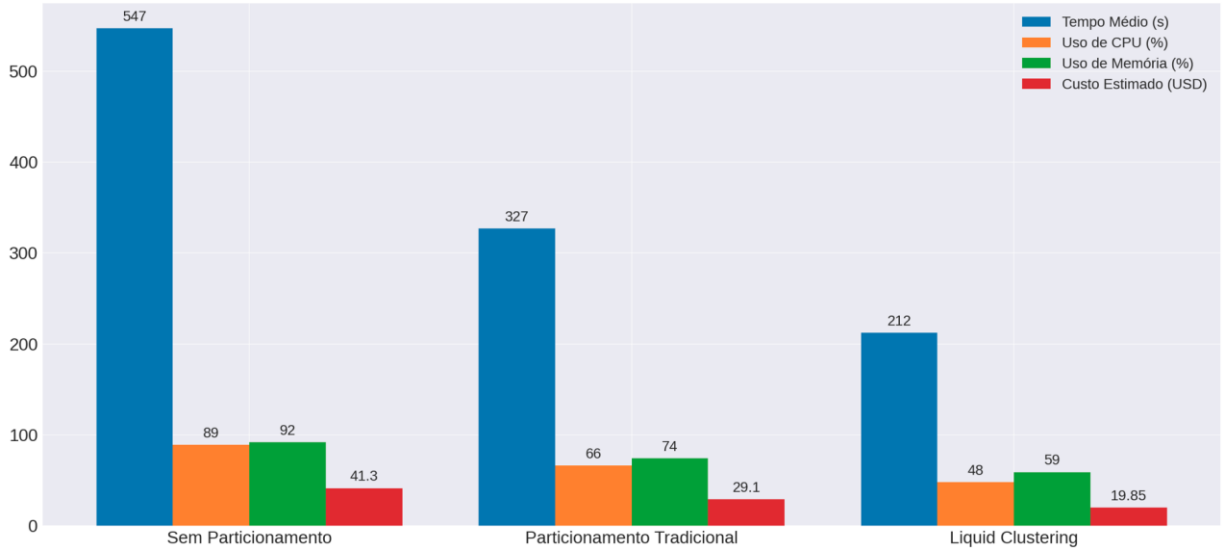
As consultas foram executadas por meio de notebooks em PySpark, simulando operações típicas de pipelines de dados em ambientes de ciência de dados e engenharia de dados. A Tabela 1 apresenta os resultados médios obtidos para cada cenário, considerando métricas de tempo de execução, uso de CPU, uso de memória e custo estimado. A execução dessas consultas gerou os resultados apresentados na Figura 2.

Tabela 1 – Comparativo de desempenho entre os três cenários de particionamento

Cenário	Tempo Médio (s)	Uso de CPU (%)	Uso de Memória (%)	Custo Estimado (USD)
Sem Particionamento	547	89	92	41,30
Particionamento Tradicional	327	66	74	29,10
Liquid Clustering	212	48	59	19,85

Fonte: os autores

Figura 2 - Comparativo de desempenho entre os três cenários de particionamento.



Fonte: os autores

No cenário sem particionamento, verificou-se o pior desempenho em todas as métricas avaliadas, com tempo médio de execução de 547 segundos, elevado consumo de CPU (89%) e memória (92%), além de maior custo operacional (USD 41,30). Esse comportamento pode ser atribuído à ausência de organização física dos dados, o que exige a realização de full scans frequentes durante a execução das consultas, aumentando significativamente o custo computacional e reduzindo a eficiência do processamento, conforme previsto por Zaharia et al. (2016) resultando em gargalos computacionais.

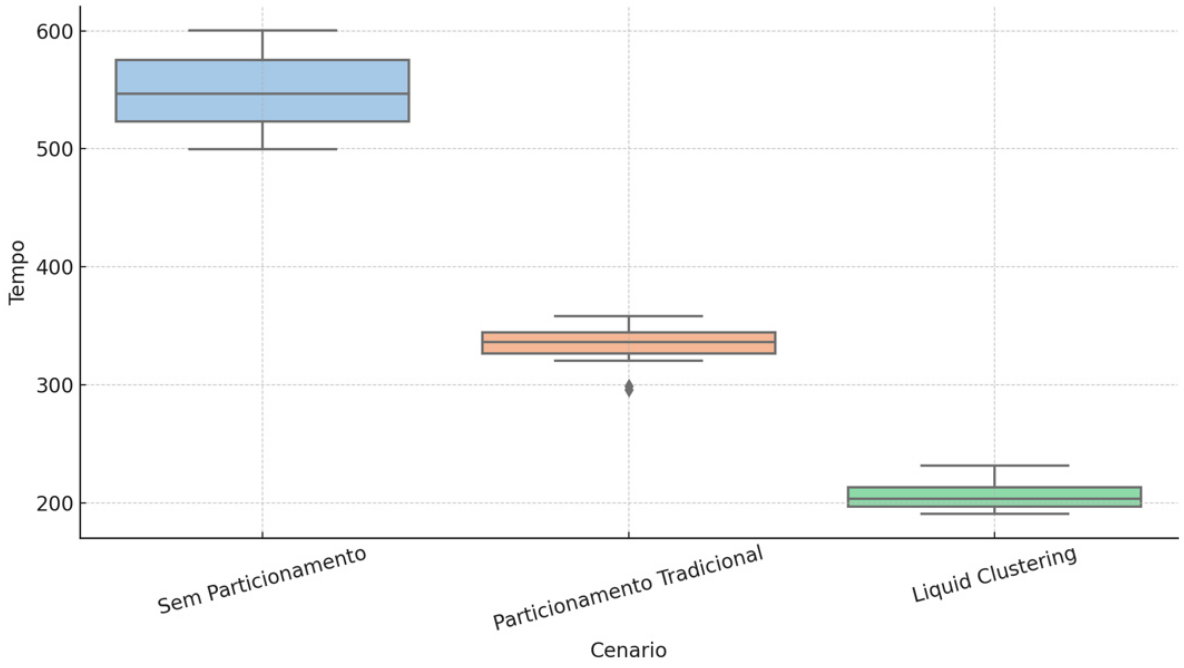
Com o particionamento tradicional, observou-se uma melhora substancial: houve redução de cerca de 40% no tempo de execução e no consumo de recursos em relação ao cenário anterior. O tempo médio foi reduzido para 327 segundos, com uso de CPU de 66% e memória de 74%, além de redução no custo para USD 29,10. Essa abordagem demonstrou eficiência em filtros sobre colunas altamente seletivas, como

UF e CNAE, mas mostrou-se menos eficaz em padrões de consulta mais dinâmicos. Esses ganhos podem ser explicados pela capacidade do particionamento em restringir o volume de dados processados durante as consultas, especialmente em operações com filtros seletivos. No entanto, essa abordagem mostrou limitações em cenários com padrões de consulta mais dinâmicos, uma vez que a estrutura de particionamento permanece estática. Isso corrobora os achados de Damji et al. (2020) que destacam a limitação de particionamentos estáticos em cargas de trabalho variáveis.

O cenário com particionamento via liquid clustering apresentou os melhores resultados em todos os aspectos, com tempo médio de execução de 212 segundos, uso de CPU de 48%, uso de memória de 59% e custo reduzido para USD 19,85. A técnica demonstrou adaptabilidade superior ao reagrupar dinamicamente os dados com base nos padrões de acesso mais frequentes. O tempo de execução caiu mais de 60% em relação ao cenário sem particionamento e cerca de 35% em relação ao particionamento tradicional. Esses resultados indicam que o liquid clustering é capaz de se adaptar dinamicamente aos padrões de acesso aos dados, reorganizando fisicamente os arquivos de forma mais eficiente. Além disso, o uso de CPU e memória foi otimizado, e o custo total da execução foi reduzido em mais de 50% — evidenciando a eficácia de técnicas adaptativas em ambientes de Big Data com cargas de consulta variadas.

Para validar estatisticamente as diferenças observadas nos tempos de execução entre os cenários, foi aplicada a ANOVA de Welch, recomendada em casos de variâncias desiguais entre grupos (heterocedasticidade), conforme verificado por meio do teste de Levene. A inspeção visual dos dados, apresentada na Figura 3, também indicou diferenças relevantes na dispersão entre os cenários.

Figura 3 – Distribuição dos Tempos por Cenário



Fonte: os autores

Antes da inferência, foram calculadas algumas medidas descritivas para os tempos de execução nos três cenários como apresentado na Tabela 2. Estas estatísticas descritivas indicam uma redução consistente não apenas na média, mas também na dispersão dos tempos de execução, especialmente no cenário com liquid clustering, que apresentou menor desvio padrão (12,44) e intervalo interquartil (11,93), evidenciando maior estabilidade e previsibilidade no desempenho.

Tabela 2 – Comparativo de desempenho entre os três cenários de particionamento

Estratégia	Média (s)	Mediana (s)	Desvio Padrão (s)	IQR (s)
Sem Particionamento	547,03	546,53	32,01	52,28
Particionamento Tradicional	335,00	336,31	16,62	22,55
Liquid Clustering	206,77	203,47	12,44	11,93

Fonte: os autores

Os dados descritivos reforçam a tendência de queda progressiva nas métricas de tempo conforme se adota estratégias de particionamento mais sofisticadas. A aplicação da ANOVA de Welch retornou valor-p < 0,001, permitindo rejeitar a hipótese nula de igualdade entre as médias e confirmando que pelo menos um dos cenários apresenta diferença estatisticamente significativa. Esses resultados confirmam a hipótese inicial do estudo: estratégias inteligentes de particionamento, como o liquid clustering, têm impacto direto e significativo na eficiência e no custo operacional de pipelines em

ambientes distribuídos. Conforme discutido por Cherukuri et al. (2021) e Shanmukha Eeti et al. (2021), a combinação entre arquitetura de dados bem planejada e técnicas adaptativas de organização é essencial para a escalabilidade e desempenho de soluções de ciência de dados em larga escala.

Do ponto de vista arquitetural, os resultados indicam a necessidade de transição de estratégias estáticas para abordagens adaptativas na organização de dados em ambientes distribuídos. Técnicas como o liquid clustering permitem alinhar dinamicamente a estrutura física dos dados aos padrões de acesso, reduzindo leituras desnecessárias e otimizando o uso de recursos computacionais. Esse comportamento está em consonância com arquiteturas modernas de dados, como o modelo lakehouse, bem como com abordagens recentes baseadas em otimização orientada por metadados e particionamento dinâmico, que visam aumentar a eficiência e a escalabilidade de pipelines de dados em larga escala (ZAHARIA et al., 2016; DAMJI et al., 2020; CHERUKURI et al., 2021; SHANMUKHA EETI et al., 2021).

5 CONCLUSÕES

Os resultados obtidos nesta pesquisa confirmam a hipótese de que a aplicação de estratégias inteligentes de particionamento de dados podem otimizar significativamente o desempenho de pipelines em ambientes distribuídos. Dentre os três cenários avaliados, o particionamento por liquid clustering, recurso nativo do Databricks, apresentou os melhores indicadores de desempenho em termos de tempo de execução, uso de CPU, uso de memória e custo financeiro, cumprindo plenamente os objetivos gerais e específicos estabelecidos.

A aplicação da ANOVA de Welch mostrou que as diferenças nos tempos de execução entre os três cenários analisados — sem particionamento, com particionamento tradicional e com liquid clustering — são estatisticamente significativas ($p < 0{,}001$), validando com rigor as observações feitas na análise descritiva.

O liquid clustering apresentou o melhor desempenho, com uma redução média de aproximadamente 61% no tempo de execução em relação ao cenário sem particionamento, e de cerca de 35% em relação ao particionamento tradicional. Essa eficiência também tem potencial para representar economia de recursos

computacionais e custos em ambientes em nuvem.

O cenário sem particionamento demonstrou ser o mais ineficiente, reforçando a importância da organização física dos dados em ambientes de Big Data. O particionamento tradicional trouxe ganhos relevantes, mas sua limitação em cenários com padrões variáveis de consulta foi superada pela adaptabilidade oferecida pelo liquid clustering.

A pesquisa demonstra, portanto, a viabilidade e a vantagem do uso de recursos pré-construídos da plataforma Databricks para resolver demandas reais do mercado por eficiência operacional e redução de custos em projetos de ciência de dados e engenharia de dados em nuvem.

Portanto, os resultados obtidos validam a aplicação do liquid clustering como solução de alto impacto para ambientes de processamento massivo, especialmente quando há variabilidade nos padrões de acesso aos dados, consolidando-se como uma prática recomendada para engenheiros, cientistas, analistas e arquitetos de dados que atuam em plataformas como o Databricks. Como possibilidade de trabalhos futuros, sugere-se expandir os testes para outros tipos de workloads, diferentes tamanhos de dataset e avaliar o impacto do liquid clustering em consultas em tempo real e em aplicações de machine learning em produção**.**

REFERÊNCIAS

ALAGARSAMY, Ram. **AI-powered query data optimization in financial systems**. ResearchGate, 2025. Disponível em: https://www.researchgate.net/publication/389749827_AI-Powered_Query_Data_Optimization_in_Financial_Systems. Acesso em: 15 maio 2025.

AUNG, Thandar; MAW, A. **Analytics of reliability for real-time big data pipeline architecture**. University of Information Technology, Yangon, Myanmar, v. 004, n. 2017, p. 42-56, abr. 2017. Disponível em:https://meral.edu.mm/record/6257/files/Analytics%20of%20Reliability%20for%20Real-Time%20Big%20Data%20Pipeline%20Architecture.pdf. Acesso em: 15 maio 2025.

BHATTACHARYA, Devipsita; CURRIM, Faiz; RAM, Sudha. **Evaluating distributed computing infrastructures: an empirical study comparing Hadoop deployments on cloud and local systems**. IEEE Transactions on Cloud Computing, v. 9, n. 3, p. 1075-1088, 1 jul. 2021. Disponível em: https://doi.org/10.1109/tcc.2019.2902377. Acesso em: 15

maio 2025.

CHERUKURI, H.; GOEL, E. L.; KUSHWAHA, G. S. **Monetizing financial data analytics: best practice**. International Journal of Computer Science and Publication, v. 11, n. 1, p. 76-87, 2021. Disponível em: https://rjpn.org/ijcspub/viewpaperforall.php?paper=IJCSP21A1011. Acesso em: 15 maio 2025.

DATABRICKS. **Delta Lake: The Definitive Guide**. Databricks, 2024. Disponível em: https://delta.io. Acesso em: 27 março 2026.

DAMJI, Jules et al. **Learning Spark: Lightning-fast data analytics**. 2. ed. Sebastopol: O'Reilly Media, 2020. Disponível em: https://www.oreilly.com/library/view/learning-spark-2nd/9781492050032/.

DARAM, S. **Impact of cloud-based automation on efficiency and cost reduction: a comparative study**. The International Journal of Engineering Research, v. 8, n. 10, p. a12-a21, 2021. Disponível em: https://tijer.org/tijer/papers/TIJER2110002.pdf. Acesso em: 15 maio 2025.

DATABRICKS. **Comprehensive guide to optimize Databricks, Spark and Delta Lake workloads**. Disponível em: https://www.databricks.com/discover/pages/optimize-data-workloads-guide. Acesso em: 15 maio 2025.

EETI, E. S.; JAIN, E. A.; GOEL, P. **Implementing data quality checks in ETL pipelines: best practices and tools**. International Journal of Computer Science and Information Technology, v. 10, n. 1, p. 31-42, 2020. Disponível em: https://rjpn.org/ijcspub/viewpaperforall.php?paper=IJCSP20B1006. Acesso em: 15 maio 2025.

KUMAR, A.; SINGH, R. **Metadata-driven optimization of distributed ETL pipelines in cloud-native data warehouses**. Journal of Data Engineering, 2024.

LI, Y. et al. **Adaptive data partitioning for distributed query processing**. arXiv preprint, 2021.

MOORE, David S. **A estatística básica e sua prática**. 9. ed. Rio de Janeiro: LTC, 2023. 626 p. Disponível em: https://www.grupogen.com.br/livro-a-estatistica-basica-e-sua-pratica-david-s-moore-william-i-notz-e-michael-a-fligner-editora-ltc-9788521638605.

PAMADI, E. V. N. **Designing efficient algorithms for MapReduce: a simplified approach**. The International Journal of Engineering Research, v. 8, n. 7, p. 23-37, 2021. Disponível em: https://tijer.org/tijer/papers/TIJER2107003.pdf. Acesso em: 15 maio 2025.

PAMADI, Vishesh Narendra; PANDEY, Priya; GOEL, Om. **Comparative analysis of optimization techniques for consistent reads in key-value stores**. International Journal of Creative Research Thoughts, v. 9, n. 10, p. d797-d813, out. 2021. Disponível em: https://www.researchgate.net/publication/388959917_Comparative_Analysis_Of_Optimization_Techniques_For_Consistent_Reads_In_Key-Value_Stores. Acesso em: 15 maio 2025.

SILVA, Adryan Felipe Marques da; BALTHAZAR, Glauber da Rocha; FERREIRA, Eduardo Augusto; SANTOS, Everton Souza dos. **Comparative analysis of blocking and non-blocking models in rest APIS**. Cuadernos de Educación y Desarrollo, [S.L.], v. 17, n. 8, p. 01-23, 5 ago. 2025. Brazilian Journals. http://dx.doi.org/10.55905/cuadv17n8-013.

SHAIKH, Eman et al. **Apache Spark: a big data processing engine**. In: IEEE Middle East and North Africa Communications Conference (MENACOMM), 2., 2019. Anais [...]. IEEE, nov. 2019. p. 1-6. Disponível em: https://doi.org/10.1109/menacomm46666.2019.8988541. Acesso em: 15 maio 2025.

SHANMUKHA, EETI; CHAURASIA, Ajay Kumar; SINGH, Tikam. **Real-time data processing: an analysis of PySpark's capabilities**. International Journal of Research and Analytical Reviews, v. 8, n. 3, p. 929-939, 2021. Disponível em: https://www.academia.edu/124656194/Real_Time_Data_Processing_An_Analysis_of_PySparks_Capabilities. Acesso em: 15 maio 2025.

YANG, Ming et al. **An efficient storage and service method for multi-source merging meteorological big data in cloud environment**. EURASIP Journal on Wireless Communications and Networking, v. 2019, n. 1, 29 out. 2019. Disponível em: https://doi.org/10.1186/s13638-019-1576-0. Acesso em: 15 maio 2025.

ZAHARIA, Matei et al. **Apache Spark: a unified engine for big data processing**. Communications of the ACM, v. 59, n. 11, p. 56-65, 2016. Disponível em: https://dl.acm.org/doi/10.1145/2934664. Acesso em:15 maio 2025.