Gerenciamento de Migração de Dados: Um Workflow Eficiente para Empresas*

Gustavo Moraes[1], **Victor Misael**[1], **Angelo Brayner**[1]

[1]Departamento de Computação – Universidade Federal do Ceará (UFC)
Campus do Pici – Fortaleza – CE – Brasil

{gustavo.moraes, victor.misael}@alu.ufc.br, brayner@dc.ufc.br

Abstract. *Data migration between different database systems (DBMS) is crucial for companies to keep up with technological advancements and optimize their processes. This article proposes a workflow to assist in conducting migrations, from planning to post-migration. The objective is to provide a roadmap for migrations, reducing execution time, mitigating risks, and ensuring data integrity and security. The proposed approach was validated in a real company scenario, demonstrating its feasibility and effectiveness.*

Resumo. *A migração de dados entre diferentes sistemas de bancos de dados (SBDs) é crucial para empresas acompanharem a evolução tecnológica e otimizarem seus processos. Este artigo propõe um workflow para auxiliar a condução de migrações, desde o planejamento até a pós-migração. O objetivo é fornecer um roteiro para migrações, reduzindo o tempo de execução, mitigando riscos e garantindo a integridade e segurança dos dados. A abordagem proposta foi validada em um cenário real de migração complexa, confirmando sua viabilidade e efetividade.*

1. Introdução

As organizações se veem diante da necessidade de migrar seus dados entre sistemas e ambientes de armazenamento devido a fatores como: constante avanço tecnológico, surgimento de novas demandas de negócio, redução de custos, busca por otimização do desempenho e da escalabilidade. No entanto, a migração de dados pode apresentar diversos desafios, requerendo uma metodologia sólida e consistente para guiar esse processo [Hussein et al. 2021, Thalheim and Wang 2013].

A migração de banco de dados é uma tarefa crítica, muitas vezes subestimada, resultando em riscos significativos para a integridade e segurança dos dados. Um dos principais desafios é a incompatibilidade de tipos de dados entre sistemas distintos, como o tipo *int unsigned* no MySQL e *int* no PostgreSQL, que não possui suporte para tipos *unsigned*. Isso exige intervenções para garantir a consistência dos dados, evitando efeitos negativos organizacionais e legais. Além disso, problemas de desempenho são comuns após a migração, impactando a eficiência operacional.

Existem diferentes tipos de migração, como a migração entre diferentes Sistemas Gerenciadores de Banco de Dados (SGBDs), entre SGBDs iguais, porém com versões diferentes, entre SGBDs com diferentes modelos de dados (por exemplo, migrar um BD

*Uma versão estendida deste artigo está disponível em https://github.com/VicMisael/IDataMig.

relacional para NoSQL) [Ellison et al. 2018] e, por fim, entre SGBDs residentes em diferentes ambientes computacionais (por exemplo, de ambiente local para uma nuvem). Portanto, a escolha do processo de migração é crucial para o sucesso da transição de banco de dados. A migração para a nuvem pode trazer benefícios como escalabilidade, custos reduzidos e maior disponibilidade, mas também apresenta desafios como questões de segurança, compatibilidade e desempenho. Existem diferentes tipos de migração a considerar: entre SGBDs diferentes, entre versões diferentes do mesmo SGBD, entre modelos de dados diferentes (por exemplo, relacional para NoSQL), e entre diferentes ambientes computacionais (por exemplo, local para a nuvem) [Ellison et al. 2018].

Este artigo apresenta uma metodologia genérica para garantir um processo de migração eficiente, tanto de ponto de vista de correção, como do aspecto tempo de execução. O objetivo é fornecer uma estratégia abrangente e sistemática, que oriente as organizações durante todo o processo de migração, desde a fase de planejamento até a fase de pós-migração. O *workflow* descrito neste trabalho fornece às organizações uma base sólida para realizar qualquer tipo de migração de bancos de dados, que mitigue riscos e assegure a integridade dos dados.

Este trabalho está organizada da seguinte forma: na Seção 2, discute-se propostas relacionadas ao processo de migração. Na Seção 3, o processo de migração proposto é descrito. Na Seção 4, a aplicação do *workflow* em um cenário empresarial real é detalhada. Por fim, a Seção 5 conclui este trabalho.

2. Trabalhos relacionados

A migração de banco de dados evolui com o avanço tecnológico, incorporando novas estratégias e ferramentas. Estudos como [Elamparithi and Anuratha 2015] analisam as estratégias de migração de bancos relacionais entre 1987 e 2008, destacando limitações e apresentando uma vasta coleção de diferentes tipos de ferramentas de migração. Um aspecto crucial desse processo é a modelagem conceitual, como discutido em [Mayr and Thalheim 2021], que desempenha um papel fundamental na compreensão e representação de dados. A capacidade de transformar modelos conceituais é essencial para garantir a preservação da semântica e a integridade dos dados durante a migração. Com o desenvolvimento de bancos de dados NoSQL, novas abordagens surgiram, incluindo a migração de bancos relacionais para modelos NoSQL. Por exemplo, [Nan and Bai 2019] propõe um algoritmo para transformar modelos de Entidade-Relacionamento (ER) em modelos de grafos, preservando integridade e restrições, enquanto [Karnitis and Arnicans 2015] oferece uma solução semiautomática para migração para bancos orientados a documentos, criando modelos lógicos refinados e *templates* personalizados.

Além do aspecto da migração envolvendo o tipo de banco de dados de origem (legado) ou destino com suas características específicas, é importante considerar os métodos de armazenamento. No contexto atual, muitas empresas optam por realizar uma transição para sistemas em nuvem. O estudo de [Ellison et al. 2018] aborda a migração de bancos de dados para a nuvem, destacando a importância de compreender a duração, os custos de migração e os custos operacionais futuros.

Esses estudos contribuem significativamente para o campo da migração de dados, relatando problemas específicos enfrentados devido às características dos bancos de da-

dos de origem ou destino e ao uso de armazenamento em nuvem. Por exemplo, o trabalho de [Wang et al. 2014] apresenta uma metodologia de migração que inclui três estratégias principais: extração dos dados do sistema legado, conversão e carregamento no destino, alinhando-se ao conceito de ETL (*Extract*, *Transform*, *Load*) [Thalheim and Wang 2013]. Esses procedimentos são essenciais para lidar com problemas específicos, como incompatibilidades de tipos de dados, garantindo uma migração mais segura e eficaz.

No entanto, é importante destacar que a migração de dados vai além de um simples procedimento ETL. O artigo de [Hussein et al. 2021] demonstra que a migração de dados é um processo multifacetado que começa com a análise dos dados legados e culmina na reconciliação dos dados carregados em novas aplicações. Este processo pode ser complexo, exigindo testes para garantir a qualidade dos dados e pode ser muito custoso se as melhores práticas não forem seguidas. O artigo propõe uma metodologia de três fases: planejamento, migração e validação.

Em síntese, os estudos sobre migração de banco de dados fornecem metodologias para enfrentar desafios na transição entre sistemas, assegurando a integridade dos dados e a eficiência do processo. Baseando-nos em trabalhos como o de [Hussein et al. 2021] e em um caso real de migração na indústria, propomos uma metodologia expandida que vai além do ETL, incorporando perspectivas teóricas e enfatizando a documentação do processo [Thalheim and Wang 2013].

3. DB-SMigra: Um *Workflow* para migração inteligente de bancos de dados

Como já afirmado anteriormente, vários cenários podem levar as empresas a recorrer ao processo de migração de bancos de dados. Contudo, esse processo apresenta riscos desafiadores. No intuito de minimizar riscos durante o processo de migração, descreveremos nesta seção o DB-SMigra, um *workflow* para conduzir o processo de migração, independente do tipo de migração.

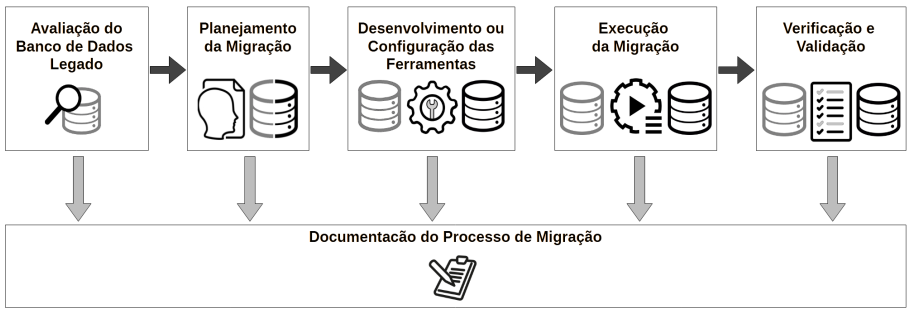

Figura 1. *Workflow* para o Processo de Migração de Bancos de Dados.

A Figura 1 apresenta o *workflow* proposto. Observe que ele é dividido em seis etapas. Cada etapa é delineada para proporcionar uma compreensão clara e uma execução eficaz do processo. Começando com a avaliação do banco de dados legado, passando pelo planejamento da migração, desenvolvimento ou configuração das ferramentas necessárias, execução da migração, até a verificação e validação final. É importante destacar que a etapa de documentação é realizada continuamente ao longo de todo o processo de migração, à medida que as atividades são desenvolvidas. Isso proporciona uma visão consolidada e contínua do *workflow*.

Além das opções de migração parcial ou completa dos dados legados e da compatibilidade dos sistemas de gerenciamento de banco de dados origem ou destino (migração

heterogênea ou homogênea), é importante considerar o número de bancos de dados envolvidos no processo. Em muitos casos, apenas um banco de dados de origem e destino está presente (1:1), mas outras possibilidades incluem a consolidação (n:1), onde vários bancos de dados legados são migrados para um único, a distribuição (1:n), onde um banco de dados legado é distribuído em vários, e a redistribuição (n:m), onde vários bancos de dados legados são redistribuídos em vários destinos [Google Cloud 2024]. Compreender qual cenário de migração se aplica facilita a adaptação das etapas (ver Figura 1) para se adequarem ao número de bancos de dados envolvidos.

Nas Seções a seguir descrevemos as etapas junto com as suas atividades, perspectivas e documentação sugerida. Vale ressaltar que as atividades não necessariamente precisam ser executadas em ordem sequencial, diversas atividades podem ser feitas em paralelo deixando o processo mais dinâmico.

3.1. Etapa 1: Avaliação do banco de dados legado

Esta etapa visa compreender a estrutura e o funcionamento do banco de dados legado. Inclui atividades como a análise inicial da modelagem do banco de dados legado, a análise da configuração física do banco de dados legado, a análise da carga de trabalho do banco de dados legado, a identificação de dados históricos ou redundantes e a detecção de inconsistências nos dados do banco de dados legado. Garantir a qualidade dos dados é crucial para a migração, pois dados corretos e limpos reduzem custos e riscos, além de apoiar decisões estratégicas [Azeroual and Jha 2021].

3.2. Etapa 2: Planejamento da migração

Nesta etapa, o foco é a elaboração de um plano de migração. Inclui atividades como a definição da configuração física e otimizações do novo banco de dados, o mapeamento entre dados de origem e destino, a escolha da abordagem de migração (única ou incremental), e a seleção de métodos ETL (*Extract*, *Transform*, *Load*) ou ELT (*Extract*, *Load*, *Transform*) para a extração de dados [Singhal and Aggarwal 2022]. Além disso, envolve o estabelecimento de metas de desempenho para o novo banco de dados, a definição de permissões de acesso e a criação de um plano de backup e reversão em caso de erros. Outras atividades incluem a avaliação do custo e impacto da migração e a elaboração de planos de testes.

Por fim, temos a avaliação de requisitos específicos de arquitetura e funcionalidades. Busque analisar as necessidades do novo ambiente de banco de dados, considerando aspectos técnicos e operacionais que afetam a eficiência, escalabilidade e robustez do sistema. Um ponto crucial é equilibrar o desempenho com a disponibilidade dos dados, decidindo entre otimizações de leitura e escrita ou estratégias de alta disponibilidade, como clusterização para distribuir carga e replicação para garantir cópias de dados em diferentes locais [Fuaad et al. 2018]. Além disso, pode ser necessário implementar técnicas como busca por similaridade ou fonéticas, usar bancos de dados de cache para melhorar o desempenho, gerenciar dados geográficos e aplicar criptografia para proteção de informações sensíveis. É importante também considerar a segurança dos dados, conformidade com regulamentos, integração com sistemas existentes e suporte a transações distribuídas. Uma avaliação abrangente desses fatores é fundamental para uma migração bem-sucedida, culminando em um relatório detalhado que justifique cada requisito e descreva seu processo de implementação incremental.

3.3. Etapa 3: Desenvolvimento ou configuração das ferramentas de migração

Esta etapa consiste no desenvolvimento ou configuração das ferramentas essenciais para realizar a migração. Inclui atividades como a configuração ou desenvolvimento da ferramenta de migração, o desenvolvimento de rotinas de testes para verificação e validação, o monitoramento de erros e métricas de desempenho, e a simulação de migração para avaliar a eficácia da ferramenta. A escolha entre configurar uma ferramenta de migração existente ou desenvolver uma nova deve considerar requisitos como a migração entre bancos de dados heterogêneos, migração paralela e incremental, reengenharia de dados, testabilidade e usabilidade com baixa curva de aprendizado [Neto et al. 2013].

3.4. Etapa 4: Execução da migração

Esta etapa visa realizar a execução controlada do plano de migração. Inclui atividades como a definição das permissões de acesso e segurança necessárias, a verificação da capacidade do sistema de destino e a execução controlada da migração com monitoramento em tempo real. Essas atividades garantem que a migração ocorra de forma segura e eficiente, minimizando riscos e interrupções nos serviços.

3.5. Etapa 5: Verificação e validação

Esta etapa é dedicada à verificação e validação dos dados migrados. Inclui atividades como a comparação entre os dados migrados e os dados legados, simulações e testes de integração com outros sistemas e a documentação dos resultados da migração. Documente a análise comparativa entre os dados migrados e os dados legados para garantir precisão e integridade. Utilize métodos estatísticos, como verificar a correspondência no número de tuplas ou instâncias, e, se necessário, compare linha a linha ou instância a instância, conforme o modelo de dados [Haller 2009].

4. Instanciação do DB-SMigra em uma migração real

Nesta seção, descrevemos um caso de uso bem-sucedido do DB-SMigra em uma empresa, onde um banco de dados MySQL foi migrado para PostgreSQL com replicação (mestre-escravo) [Ozsu and Valduriez 2001]. A migração envolveu 120 tabelas e cerca de 460 índices, totalizando aproximadamente 30 milhões de linhas e 5,60 GB. Cerca de 85% dos dados estavam concentrados em uma única tabela. Após uma limpeza pré-migração, duas tabelas obsoletas foram excluídas, e não foram detectadas inconsistências nos dados restantes.

O novo ambiente de armazenamento foi configurado em um servidor local dedicado, facilitando a migração entre os modelos de dados, pois ambos são relacionais. As alterações foram mínimas, concentrando-se principalmente em transformações de tipos de dados. A migração foi dividida em duas abordagens: uma migração única para a maioria das tabelas e uma migração incremental para a tabela com o maior volume de dados. A prioridade era garantir a alta disponibilidade do sistema. Para isso, foi usado o Patroni[1], que gerencia um banco de dados primário com duas réplicas. Em caso de falha do banco de dados primário, uma das réplicas pode assumir o controle. Requisitos específicos para armazenamento de dados geográficos foram atendidos com a extensão PostGIS[2].

[1]https://patroni.readthedocs.io/en/latest/

[2]https://postgis.net/

Apesar da existência de ferramentas para migração de MySQL para PostgreSQL, optamos por desenvolver uma ferramenta própria para automatizar várias atividades da metodologia de migração. Entre essas atividades, destacam-se a detecção de inconsistências, o mapeamento entre origem e destino, o monitoramento de erros, a reversão em caso de falhas, a transferência de dados e as verificações de integridade. O código da ferramenta, juntamente com diversos experimentos de desempenho, está disponível em um repositório online[3]. Embora, essa decisão tenha sido mais demorada, proporcionou maior controle do código, especialmente para a implementação de conversores de tipos de dados, como para dados geográficos.

A ferramenta desenvolvida lê e converte os esquemas das tabelas legadas para o novo ambiente, migra tuplas, chaves primárias, regras de integridade, chaves estrangeiras, índices e sequências (tipo *auto-increment*). Também inclui um sistema de *log* que registra todo o processo e um sistema de tolerância a falhas que permite reverter as transações e continuar a partir de um ponto estável em caso de problemas. Durante a execução, todas as tabelas foram inicialmente migradas, exceto a tabela com o maior volume de dados, que foi migrada de forma incremental. Diversas simulações com dados de teste foram realizadas para garantir a eficácia do processo. O tempo total de migração do banco de dados legado foi de aproximadamente 45 minutos, enquanto o processo completo, desde o planejamento até a pós-migração, levou cerca de 35 dias úteis.

Um dos pontos-chave para o sucesso da migração foi a abrangência e o nível de detalhes do *workflow*, aliados ao forte incentivo à documentação de todo o processo. Isso permitiu que todas as partes interessadas no projeto contribuíssem de maneira mais eficiente, garantindo a transparência necessária para uma operação crítica e minimizando os riscos aos negócios da empresa.

5. Conclusão

Este trabalho apresentou o DB-SMigra, um *workflow* genérico para a migração de bancos de dados, com boas práticas para garantir um processo eficiente e seguro. DB-SMigra foi validado na migração em uma empresa, que queria migrar um banco de dados MySQL para o PostgreSQL com replicação (mestre-escravo).

Por fim, deve-se destacar que o sucesso de uma migração de dados depende do planejamento adequado, do uso de ferramentas apropriadas e do monitoramento contínuo para identificar e resolver possíveis problemas de forma ágil. A documentação das etapas e dos resultados da migração proporciona maior transparência e rastreabilidade do processo, permitindo que futuras migrações na empresa tirem proveito das lições aprendidas. A experiência adquirida e os métodos aplicados neste estudo servem como um guia para profissionais da área, reforçando a importância de uma abordagem estruturada.

Referências

Azeroual, O. and Jha, M. (2021). Without data quality, there is no data migration. *Big Data and Cognitive Computing*, 5(2):24.

Elamparithi, M. and Anuratha, V. (2015). A review on database migration strategies, techniques and tools. *World Journal of Computer Application and Technology*, 3(3):41–48.

[3]https://github.com/VicMisael/IDataMig

Ellison, M., Calinescu, R., and Paige, R. F. (2018). Evaluating cloud database migration options using workload models. *Journal of Cloud Computing*, 7:1–18.

Fuaad, H., Ibrahim, A., Majed, A., and Asem, A. (2018). A survey on distributed database fragmentation allocation and replication algorithms. *Current Journal of Applied Science and Technology*, 27(2):1–12.

Google Cloud (2024). Migração de banco de dados: conceitos e princípios (parte 1). `https://cloud.google.com/architecture/database-migration-concepts-principles-part-1?hl=pt-br`. Acessado em: 2024-05-29.

Haller, K. (2009). Towards the industrialization of data migration: concepts and patterns for standard software implementation projects. In *Advanced Information Systems Engineering: 21st International Conference, CAiSE 2009, Amsterdam, The Netherlands, June 8-12, 2009. Proceedings 21*, pages 63–78. Springer.

Hussein, A. A. et al. (2021). Data migration need, strategy, challenges, methodology, categories, risks, uses with cloud computing, and improvements in its using with cloud using suggested proposed model (dmig 1). *Journal of Information Security*, 12(01):79.

Karnitis, G. and Arnicans, G. (2015). Migration of relational database to document-oriented database: Structure denormalization and data transformation. In *2015 7th international conference on computational intelligence, communication systems and networks*, pages 113–118. IEEE.

Mayr, H. C. and Thalheim, B. (2021). The triptych of conceptual modeling: A framework for a better understanding of conceptual modeling. *Software and Systems Modeling*, 20(1):7–24.

Nan, Z. and Bai, X. (2019). The study on data migration from relational database to graph database. In *Journal of Physics: Conference Series*, volume 1345, page 022061. IOP Publishing.

Neto, P. S., Neto, J. R., Júnior, F. R., and Oliveira, P. (2013). Requisitos para ferramentas de migração de dados. In *Anais do IX Simpósio Brasileiro de Sistemas de Informação*, pages 887–898, Porto Alegre, RS, Brasil. SBC.

Ozsu, M. T. and Valduriez, P. (2001). *Principios de sistemas da bancos de dados distribuidos*. Campus.

Singhal, B. and Aggarwal, A. (2022). Etl, elt and reverse etl: a business case study. In *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, pages 1–4. IEEE.

Thalheim, B. and Wang, Q. (2013). Data migration: A theoretical perspective. *Data & Knowledge Engineering*, 87:260–278.

Wang, G., Jia, Z., and Xue, M. (2014). Data migration model and algorithm between heterogeneous databases based on web service. *Journal of Networks*, 9(11):3127.