

CURSO DE SISTEMAS DE INFORMAÇÃO – TCC ACADÊMICO		
() PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2020/2

INTEGRAÇÃO DE DADOS DE SISTEMA LEGADO COBOL PARA UM BANCO DE DADOS SGDB

Luiz Carlos Burigo

Prof. Gilvan Justino – Orientador

1 INTRODUÇÃO

A linguagem Cobol (Common Business Oriented Language) foi criada em 1959 em uma iniciativa do Pentágono (USA) que objetivava estabelecer uma linguagem comum de programação para ser utilizada na solução de problemas comerciais (BASTOS, 1983). Atualmente a linguagem Cobol é responsável por 43% dos sistemas bancários no EUA sendo utilizada ao menos em três agências governamentais americanas (SOPRANA, 2020).

Entre os diferenciais da linguagem Cobol pode-se mencionar sua performance e custo-benefício, alcançados devido utilização de arquivos para armazenamento de dados (BASTOS, 1983). A organização dos arquivos de dados Cobol é feita por uma sequência não estruturada de bytes, onde o sistema de arquivos não impõe nenhuma estrutura lógica para os dados, sendo responsabilidade da aplicação definir toda a sua organização (LEITE, 2013).

Nos dias atuais, uma vantagem tão importante quanto armazenar dados é conseguir através dos mesmos gerar informação valiosa por meio de ferramentas, tal como Analytics, que permite fornecer subsídios para tomada de decisão. Através de Analytics é possível realizar análises preditivas e descobrir tendências futuras, permitindo às empresas reagirem conforme ocorrem as mudanças no mercado bem como se anteciparem com relação ao seu próprio futuro (EFRAIM TURBAN, 2017). Para que isso seja possível, é necessário que os dados sejam de fácil acesso para que estejam disponíveis para serem utilizados em diversas ferramentas.

Infelizmente, os arquivos de armazenamento de dados Cobol podem ser manipulados apenas por aplicações Cobol restringindo o acesso aos dados por outras aplicações. Diante deste cenário, este trabalho apresenta a proposta de implementação de um protótipo que aborde uma solução ETL. Será feito a extração de dados mantidos por uma aplicação Cobol para migrar os dados para um cluster de banco de dados novo que permita a integração das mais diversas tecnologias. Um processo de ETL pode lidar com os requisitos complexos do processo de migração de dados, gerenciando de dados em um novo formato que satisfaça os requisitos operacionais e analíticos do negócio (ASTERA, 2020).

1.1 OBJETIVOS

O objetivo geral deste projeto é disponibilizar um protótipo para a integração de dados mantidos por uma aplicação Cobol para um novo modelo de banco de dados NoSQL em um cluster de banco de dados. Os objetivos específicos são:

- a) **manter temporariamente** os dados a serem migrados em arquivos temporários;
- b) **modelar a arquitetura** a ser implementada para migração de dados;
- c) disponibilizar programas Cobol que farão a extração de dados;
- d) definir e modelar as novas coleções na cloud em um banco de dados MongoDB;
- f) **eleger** uma tecnologia para sincronização dos dados com a nova base de dados na cloud.

2 TRABALHOS CORRELATOS

Foram observados três trabalhos correlatos que apresentam propostas semelhantes ao projeto proposto. Na subseção 2.1, Jacinto (2006) propôs uma ferramenta de extração de dados de um banco de dados restritos a aplicação COBOL através de comandos de SQL. Na subseção 2.2, Borchardt (2016) construiu uma API que disponibiliza as informações do sistema legado através de uma web service em Python com uso do framework Django. A subseção 2.3 será abordada uma aplicação desenvolvida por Gonçalves et al. (2019) que buscou a extração e migração de dados para um novo banco de dados para MongoDB. Esta aplicação lê os dados de um banco DB2 conforme query previamente definida pelo usuário e migra os dados lidos para um banco de dados MongoDB.

2.1 COBOL QUERY 1.0 – FERRAMENTA PARA EXTRAÇÃO DE DADOS EM ARQUIVOS COBOL

A gravação de dados em arquivos é uma das formas utilizadas por programas COBOL para armazenamento de dados, neste tipo de armazenamento a leitura dos dados só pode ser feita unicamente por uma aplicação COBOL. Neste sentido a forma como será feito o acesso aos dados está determinada na aplicação COBOL, para executar um acesso para leitura diferente da prevista na aplicação COBOL é necessário implementar na aplicação COBOL uma nova forma de acesso para leitura (JACINTO, 2006).

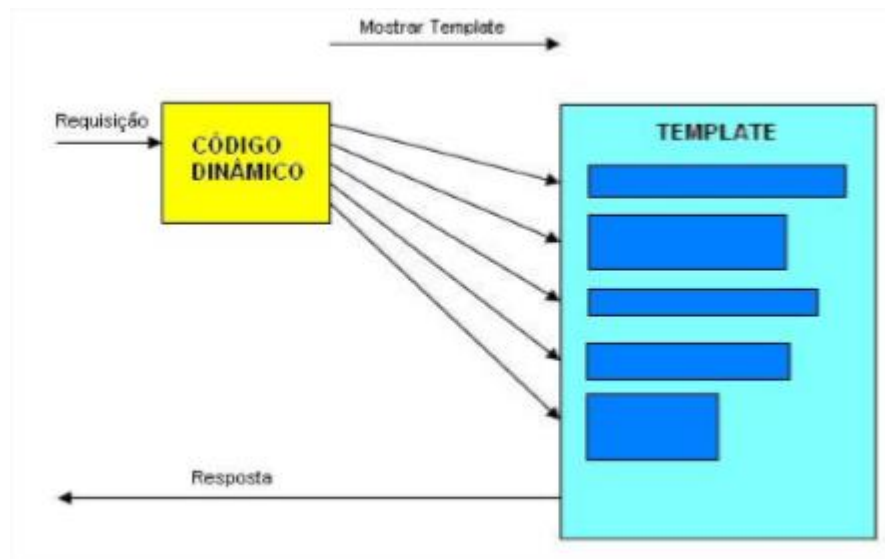
Para permitir que outras aplicações acessassem dados mantidos por aplicações Cobol, Jacinto (2006) desenvolveu uma ferramenta que possibilita ao usuário a utilização de comandos DDL (Data Definition Language). Estes comandos são enviados a um interpretador que faz uma análise léxica, sintática e semântica dos comandos. A partir desta análise a

ferramenta gera um código COBOL correspondente ao comando DDL submetido (JACINTO, 2006). O código COBOL gerado é executado pela ferramenta e faz a consulta no arquivo COBOL e por fim apresenta os dados solicitados pelo usuário.

O autor utilizou a linguagem de programação Java para desenvolvimento do projeto implementando um modelo de templates (figura 1). O programa é instanciado por uma requisição com os comandos de leitura e a ferramenta associa os valores das variáveis aos blocos de templates utilizados de acordo com as palavras reservadas utilizadas e em seguida é apresentada a resposta ao usuário (JACINTO, 2006).



Figura 1 – Fluxo da Ferramenta

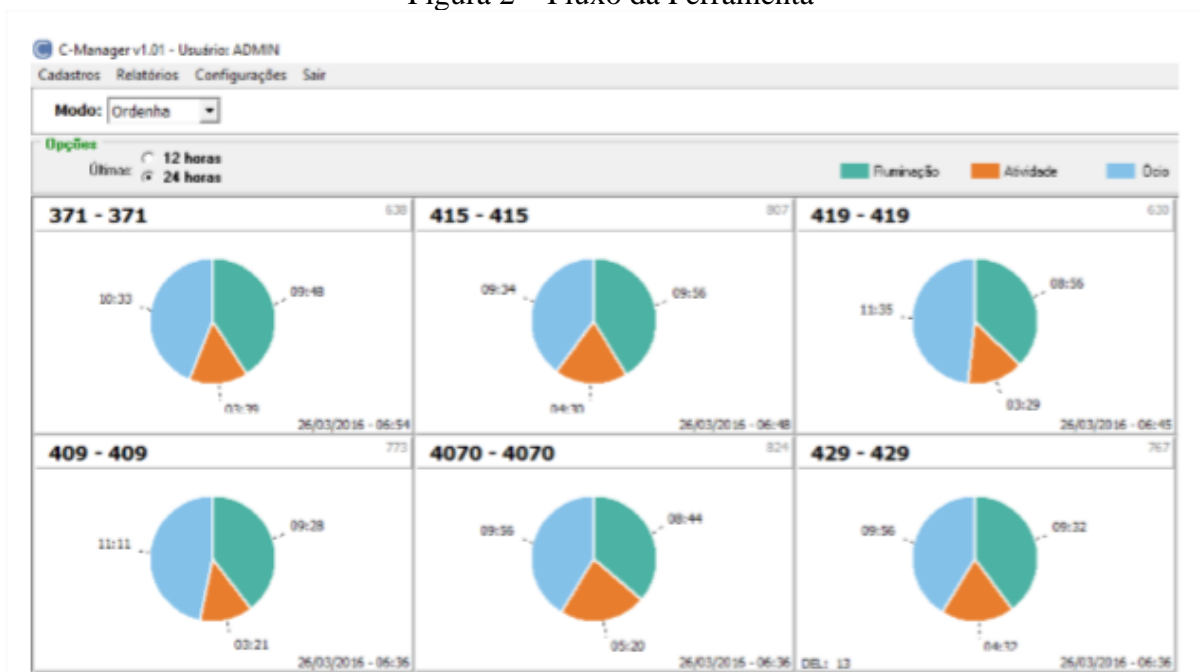


Fonte: Adaptado de Rocha (2005).

2.2 MELHORIA DE SISTEMA LEGADO DO SETOR AGROPECUARIO UTILIZANDO API

A empresa Chip Inside é desenvolvedora de um sistema agropecuário conhecido como C-Tech, que é utilizado para tratar do conforto animal e detecção de calor para gado leiteiro. O sistema foi desenvolvido em Delphi em sua versão 7 usando programação estruturada. Muitos de seus componentes não são mais atualizados nessa versão o que causava dificuldades para manutenção e implementação de novas funcionalidades, além de sua interface apresentar um padrão já ultrapassado e antiquado para as necessidades (figura 2).

Figura 2 – Fluxo da Ferramenta



Fonte: adaptado de Borchardt (2016).

Para modernizar o sistema Chip Inside, foi construída uma API que disponibiliza informações para tomada de decisão da fazenda. Com isso foi possível realizar a integração de com tecnologias, tais como aplicações web ou mobile que consumiam dados desta API. O autor adotou como estratégia o mapeamento do banco de dados existente através de funcionalidades nativas no framework Django, onde é possível através de um comando identificar a estrutura do banco utilizado (com seus atributos e relacionamentos) do sistema legado e instanciar uma classe denominada Model (BORCHARDT, 2016).

Borchardt (2016) cita como resultado a transição de dados de forma segura e confiável, uma vantagem proporcionada ao utilizar o Django Rest Framework, pois já vem implementada a autenticação bastando apenas indicar nas configurações o tipo de autenticação. Segundo Borchardt (2016) a API permitiu maior conectividade com outros sistemas e integração de dados como outras aplicações e levou novas tecnologias ao campo onde praticamente todos os grandes avanços são voltados apenas ao maquinário agrícola. Um dos aspectos apontados por Borchardt (2016) foi melhorar o desempenho da API.

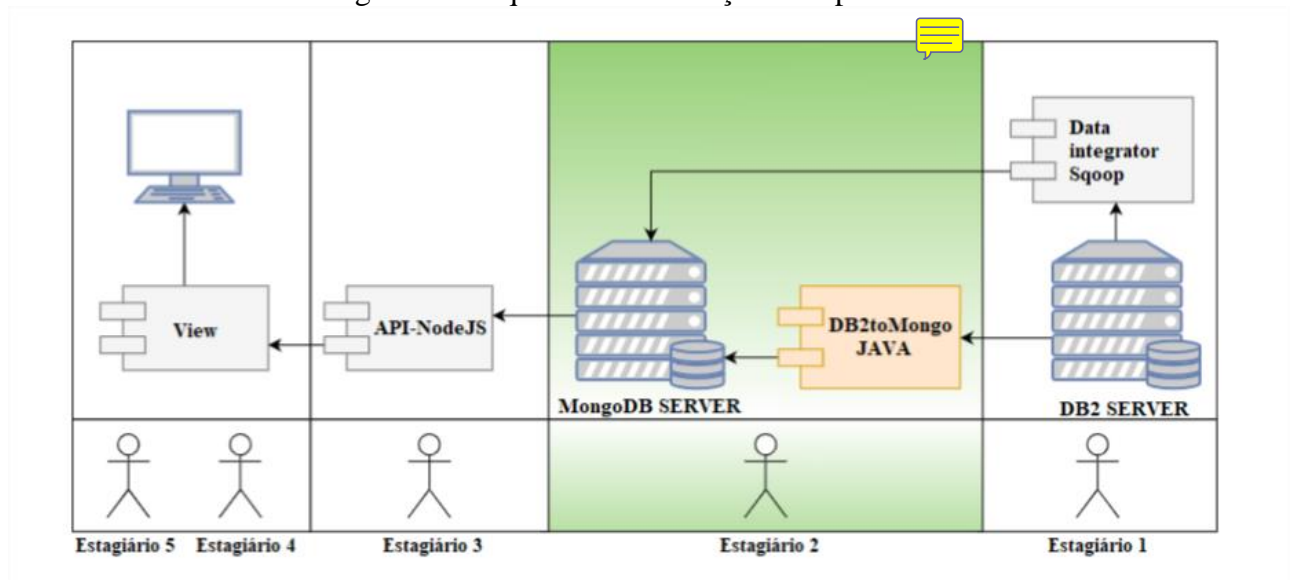
2.3 UMA APLICAÇÃO PARA MIGRAÇÃO DE DADOS DE BANCO RELACIONAL PARA MONGODB

Machado *et al.* (2019) trabalharam na construção de um software que sofria com problemas de performance devido o volume excessivo de dados. Para mitigar o problema

Machado *et al.* (2019) trouxeram como proposta a migração de dados do banco legado para uma nova estrutura em MongoDB e um banco NoSQL que no contexto do problema apresentado teria apenas função para leitura.

Sua API é um web service consumido através de web request permitindo integração tanto com outras aplicações web como com aplicações mobiles. Conforme Machado *et al.* (2019) proveram a integração a novas tecnologias, buscando auto performance para consultas de grande volume de dados e disponibilizar de forma consistente e rápida para ser consumida por outra API em Node JS. Machado *et al.* (2019) apresentaram como características a migração de dados para uma estrutura moderna de banco de dados. Os autores buscaram dividir o banco de dados em um banco de dados para escrita onde seria consistida a regra de negócio (banco legado) e uma migração da informação para um outro banco NoSQL para leitura. Outra característica encontrada apenas em Machado *et al.* (2019) foi a atualização de dados em tempo real conforme query previamente definida para fazer o insert, update e delete em real time na base de leitura. A figura 3, apresenta uma visão geral da arquitetura desenvolvida por Machado *et al.* (2019).

Figura 3 – Arquitetura da Solução Completa



Fonte: Machado *et al.* (2016).

3 PROPOSTA DO PROTÓTIPO

No decorrer desta seção serão apresentadas as justificativas para a realização do protótipo proposto, que se encontra na subseção 3.1. Serão descritos os principais requisitos do

protótipo na subseção 3.2, bem como a metodologia e o cronograma planejado para o desenvolvimento do trabalho na subseção 3.3.

3.1 JUSTIFICATIVA

Em abril de 2020, houve nos EUA uma corrida sem precedentes à procura de desenvolvedores Cobol 15, linguagem com 220 bilhões de linhas de código em uso hoje. Isto aconteceu pois era preciso efetuar a atualização de sistemas financeiros legados em Cobol para elegibilidade de pagamento de desemprego em um prazo muito curto (SOPRANA, 2020). Esta notícia denota a importância que uma linguagem legada tem para o sistema financeiro governamental americano. Mesmo nos dias atuais, a necessidade de acrescentar novas funcionalidades. Neste sentido encontram-se vários obstáculos como falta de profissionais para desenvolvimento COBOL e obsolescência da linguagem COBOL. É necessário buscar a integração de dados, funções de sistemas legados a novas tecnologias e também trazer para o mercado de trabalho condições de encontrar a oferta necessária de profissionais para atuação com tecnologias que possam integrar a sistemas legados.

Pode-se concluir, a partir do Quadro 1, que todos os trabalhos correlatos citados possuem uma convergência em relação a propor maneiras de acesso aos dados mantidos por sistemas legados assim como apresentar uma melhoria ao sistema.

Quadro 1 –Comparativo entre os trabalhos correlatos

Correlatos Características	Luiz Jacinto (2006)	Fábio Ivan Borchardt (2016)	Machado <i>et al.</i> (2019)
Migração de dados para uma estrutura moderna de banco de dados	Não	Não	Sim
Atualização de dados em real time na nova estrutura	Não	Não	Sim
Prover acesso de outras aplicações aos dados do sistema legado	Sim	Sim	Sim
Melhoria de sistema legado	Sim	Sim	Sim
Prover integração do sistema legado a novas tecnologias	Não	Sim	Sim

Fonte: elaborado pelo autor

Conforme Jacinto (2006), o mesmo possibilitou ao usuário um acesso direto ao banco dados do sistema legado sem a necessidade de migração de dados. Borchardt (2016) proveu o acesso de novas tecnologias aos dados do banco do sistema legado e também implementou

uma estrutura que possibilita a integração de novas tecnologias para disponibilizar novos serviços. Já Machado *et al.* (2019), buscaram fornecer acesso aos dados de sistemas legados por outras tecnologias bem como manter uma estrutura performática de extração de dados visando a necessidade de manipular grandes volumes de dados.

A característica de integração de sistemas legados a novas tecnologias é presente em Borchardt (2016). Com base nas características apresentadas no quadro 1, é possível identificar semelhanças entre os trabalhos correlatos e o trabalho aqui proposto. A proposta tem como objetivo migrar todos os dados de um banco de dados de acesso restrito a aplicação Cobol para um SGDB. Pretende-se modelar a arquitetura de dados de forma a ter uma base de dados performática e possibilitar efetuar consultas de dados de forma flexível. Também busca-se prover a integração de novas tecnologias a fim de possibilitar o desenvolvimento de novos serviços em tecnologias mais atuais. Diante do exposto o trabalho pode ser relevante não apenas por migrar dados, mas:

- a) permitir a segmentação de funcionalidades do sistema legado a novos serviços em tecnologias mais atuais;
- b) oferecer uma base de dados consistente para integração com ferramentas de Business Intelligence;
- c) prover cenários que colaborem com processo de melhoria contínua do sistema;
- d) conseguir integrar o sistema legado a novas tecnologias sem ter que causar indisponibilidade do mesmo para implementação de novas tecnologias.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Nesta seção serão abordados os principais Requisitos Funcionais (RF), bem como os principais Requisitos Não Funcionais (RNF). O protótipo proposto neste trabalho deverá:

- a) disparar uma trigger para realizar gravação em arquivo COBOL (RF);
- b) disparar uma trigger para realizar update em arquivo COBOL (RF);
- c) disparar uma trigger para realizar delete em arquivo COBOL (RF);
- d) realizar a carga inicial com os dados do sistema legado no banco de dados na cloud (RF);
- e) registrar em arquivos json manipulação ocorrida em arquivos COBOL (RF);
- f) enviar o arquivo json com os dados do respectivo arquivo gravado para uma aplicação sincronizadora na cloud (RF);
- g) evitar queda de desempenho do sistema por causa da migração de dados (RNF);
- h) o programa de sincronização de dados deve validar qual ação realizará (insert,

update ou delete) (RNF);

- i) o programa de sincronização deverá gravar as informações recebidas em um cluster de MongoDB na cloud (RNF).

3.3 METODOLOGIA

O protótipo será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: realizar uma revisão mais aprofundada sobre arquitetura de dados e de software para entendimento da melhor abordagem para integração de sistemas legados;
- b) levantamento de requisitos: reavaliar os requisitos funcionais e não funcionais pré-definidos e, se necessário, especificar alterações a partir das necessidades observadas durante a revisão bibliográfica;
- c) especificação e análise: prototipar as ideias antes da etapa de desenvolvimento fazendo uso de fluxograma, diagramas e UML utilizando a ferramenta online Draw Io;
- d) implementação do protótipo: implementação das triggers no Cobol que serão responsáveis na transferência de dados e extração de dados, utilização de scripts em PYTHON para monitoramento de arquivos criados para envio para cloud e gravação de dados no novo banco de dados em Mongo DB. Será utilizado a IDE Open Cobol e VS Code assim como Mongo DB Compass Community para implementação do banco de dados;
- e) verificação e validação: validar consistência de dados e performance de gravação e consulta.

As etapas serão realizadas nos períodos relacionados no quadro 2.

Quadro 2 - Cronograma

Etapas / Quinzenas	ano									
	fev.		mar.		abr.		mai.		jun.	
	1	2	1	2	1	2	1	2	1	2
Levantamento bibliográfico										
Levantamento de requisitos										
Especificação e análise										
Implementação do protótipo										
Verificação e validação										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Nesta seção são apresentados os assuntos principais que irão servir como base para construção do protótipo. A organização está disposta respectivamente desta forma: 4.1 conceitua os principais arquivos responsáveis pelo armazenamento de dados restritos a aplicação COBOL; na subseção 4.2 contextualiza as características e a importância dos sistemas legados para suas cooperações na atualidade e qual a importância da integração a novas tecnologias.

4.1 ESTRUTURA DE ARQUIVOS DE DADOS E ACESSO A DADOS PERSISTIDOS EM COBOL

Gravação de dados em arquivos é uma das formas utilizadas por programas COBOL para armazenamento de dados. Cada arquivo possui no código fonte COBOL a representação de sua estrutura com seus atributos e respectiva tipagem. Os arquivos armazenam um conjunto de dados relacionados como por exemplo um arquivo de clientes ao qual mantém os registros de todos os clientes.

Existem duas maneiras de acessar arquivos mantidos pela linguagem COBOL: o acesso sequencial e o acesso direto ou randômico (NEWCOMER, 1985). Arquivos sequenciais são aqueles que mantêm seus registros de dados distribuídos ordenados na sequência de gravação dos dados. O arquivo sequencial pode ser também ordenado desde que na sua estrutura seja implementado layout de dados para o registro, assim como uma chave que será responsável tanto pela ordenação lógica quanto física (SCHREINER, 2017).

Os arquivos podem ser organizados de forma sequencial e indexada, na organização sequencial os registros são armazenados um após o outro e são lidos na mesma ordem que são acrescentados no arquivo. As leituras destes arquivos são feitas unicamente na forma sequencial (SCHREINER, 2017). Quando há um grande volume de dados armazenados, os métodos de armazenamento sequencial com acessos aleatórios podem deixar o sistema lento. Para o armazenamento de grandes volumes de dados onde é necessário um acesso direto, indica-se o armazenamento em arquivos sequenciais indexados (SCHREINER, 2017).

Schreiner (2017) explica que um arquivo sequencial indexado contém em sua estrutura 3 áreas: uma área primária, reservada para os registros de dados ordenados por uma chave, uma área de índices, e a área de excedentes, reservada para o acréscimo de novos registros que não podem ser colocados na área principal. Neste sentido Schreiner (2017) cita como

características dos arquivos sequenciais indexados um acesso aleatório satisfatório, facilidade nas operações de busca, inserção e exclusão.

4.2 INTEGRAÇÃO DE SISTEMAS LEGADOS E NOVAS TECNOLOGIAS

O mundo da tecnologia da informação está em constante evolução. A cada dia novas tecnologias viram tendências e assim fragmentando seu uso em diversas aplicações corporativas que acabam denotando um certo problema, a falta de integração com sistemas legados (CUMMINS, 2002).

Há algumas características que demonstram que um sistema possa ser legado como: falta de documentação, tecnologia ultrapassada, sistemas grandes e complexos e que apresentam elevados números de erros e falhas (CHAVES, 2004). Apesar de todos os problemas que um sistema legado possa ter a solução ideal está no reaproveitamento do que já existe, realizando melhorias, modificações de componentes e melhoria de interface (BISPAL, 1999). Neste sentido uma solução que permita o reaproveitamento do sistema legado é a integração de sistemas, porque além de estender a vida útil do sistema legado reaproveita o que já existe (BISPAL, 1999).

A integração de sistemas tem como definição a colaboração de mais de um sistema em prol de um objetivo. Esta integração está presente tanto em sistemas robustos como sistemas ERP como em sistemas desenvolvidos em tecnologias diferentes (SOMMERVILLE e MELNIKOFF, 2003). Para a integração de sistemas ter sucesso é preciso existir a interoperabilidade entre os mesmos e assim efetuar a troca de informação de maneira eficaz (SOMMERVILLE e MELNIKOFF, 2003).

4.3 MIGRAÇÃO DE DADOS

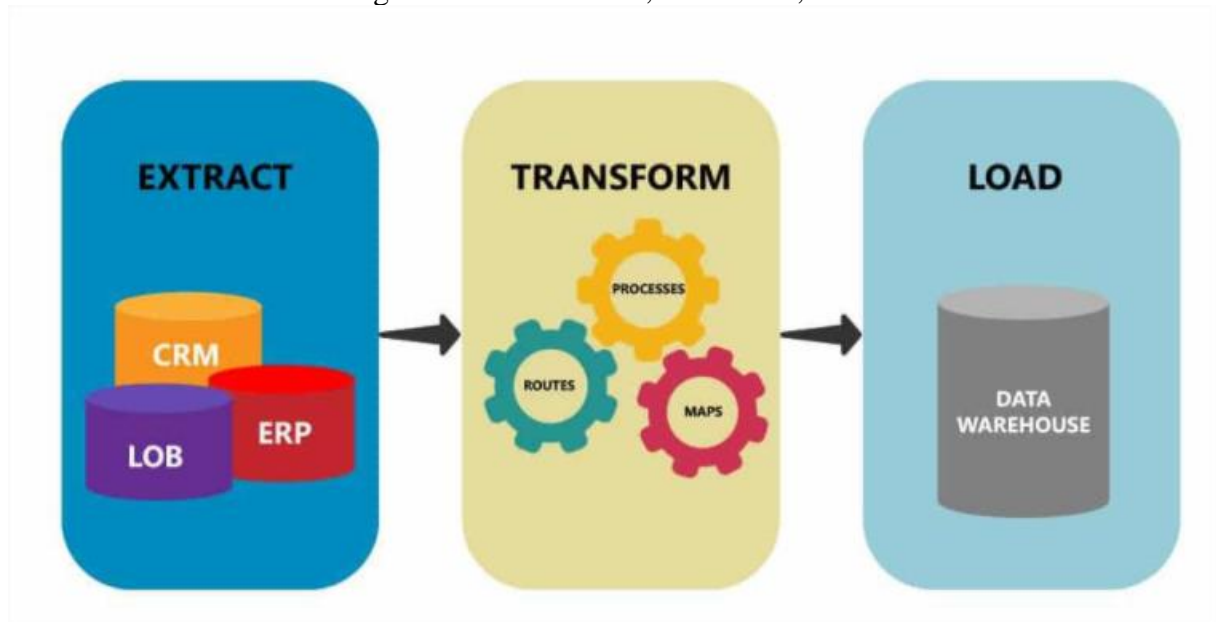
O processo de transferir dados de um sistema para outro é chamado de migração de dados. Há vários motivos que possam levar a necessidade de um projeto de migração de dados tais como substituir sistemas legados, integração com tecnologias de ponta, plano de recuperação de desastres ou até mesmo transferir dados para nuvem, eliminando o custo da infraestrutura de TI local (NAEEM, 2020). Em todos os casos os dados devem ser migrados das aplicações existentes para o novo sistema (NAEEM, 2020).

Existem vários fatores que determinam a melhor abordagem técnica para migração de dados, como requisitos de sucesso pode-se citar a combinação da manutenção da integridade dos dados, eficiência e impacto mínimo aos usuários e processos de negócios, entre as várias abordagens técnicas que temos no mercado pode-se citar o método ETL (NAEEM, 2020).

O método ETL está condicionado a um processo com três etapas, extração, transformação (figura 4) e carregamento. ETL é um método que busca padronizar a migração de dados de um banco de dados e carregar os mesmos dados em outro banco de dados após verificações de transformação em qualidade (FATIMA, 2020).

A primeira etapa do método ELT é extrair dados de uma fonte de dados de uma fonte de dados, geralmente os dados são extraídos de diversas fontes, como bancos de dados locais e na nuvem, aplicativos corporativos e sistemas de arquivos (FATIMA, 2020). A segunda etapa transformação, os dados extraídos são convertidos para o formato do novo banco de dados e então na terceira etapa é feito o carregamento, o procedimento de gravar ou empilhar dados no banco de dados destino (FATIMA, 2020).

Figura 4 – ETL Extract, Transform, Load



Fonte: Fatima (2020).

A migração de dados não apenas é um processo para mover dados de um banco de dados para outro. A migração de dados envolve grandes volumes de dados, sistemas heterogêneos, estruturas díspares e mapeamentos complexos (NAEEM, 2020). “É um processo amplamente usado de movimentação de dados, e quase todas as organizações o realizam em algum momento.” (NAEEM, 2020).

REFERÊNCIAS

BASTOS, Alex C. **Programação COBOL**, 4ª edição. Rio de Janeiro: Livros Técnicos Científicos, 1988.

Bispal, J. *et al.* **Legacy Information System: Issues and Directions**. Dublin, [1999]. disponível em: <https://www.computer.org/csdl/magazine/so/1999/05/s5103/13rRUwInv2x>. Acesso em: 28 set. 2020.

BORCHARDT, Fábio Ivan. **Melhoria de Sistema Legado do Setor Agropecuário Utilizando API**. 2016. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Centro de Ciências Exatas e Naturais, Antonio Meneghetti Faculdade, Restinga Seca.

CUMMINS, Fred A. **Integração de Sistemas: arquiteturas para integração de Sistemas e Aplicações Corporativas**. Rio de Janeiro: Campus, 2015.

JACINTO, Luiz. **Cobol Query 1.0 – Ferramenta para Extração de Dados em Arquivos COBOL**. 2006. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

LEITE, Andreza. **Sistemas de arquivos**. Florianópolis, [2013]. Disponível em: <http://docente.ifsc.edu.br/alex.forghieri/MaterialDidatico/Sistemas%20Operacionais/Material%20das%20aulas/06%20-%202024-06-2016/Sistema%20De%20Arquivos.pdf>. Acesso em: 20 set. 2020.

MACHADO, Rodrigo; GIRARDON, Gustavo; COSTA, Victor; BERNADINO, Maicon; JACOMÉ Marcos; GONÇALVES, Robson. **Uma Aplicação para Migração de Dados de Banco Relacional Para MongoDB**. Porto Alegre, [2019]. Disponível em: <https://sol.sbc.org.br/index.php/eres/article/view/8503>. Acesso em: 30 set. 2020.

NAEEM, Irfan Ahmedtehreem. **Software de Migração de Dados**. Santa Clara, nov. 2020. Disponível em: <https://www.astera.com/pt/type/blog/data-migration-software/>. Acesso em: 10 nov. 2020.

Fatima, Irfan Ahmednida. **ETL vs ELT: Qual a diferença**. Santa Clara, nov. 2020. Disponível em: <https://www.astera.com/pt/type/blog/etl-vs-elt-whats-the-difference/>. Acesso em: 27 nov. 2020.

NEWCOMER, Laurence R. **Cobol estruturado**. Tradução Lars Gustav Erik Unonius. São Paulo: McGraw-Hill do Brasil, 1985.

RICCIO, Edison L. *et al.* **Sistemas Legados e a Aplicação de Processo de Reengenharia de Software**. Em: 1ST INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT, 2004, São Paulo. Disponível em: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S1807-17752004000100008

SCHEREINER, Marcos Antonio. **Arquivo Sequencial e Indexado**. Curitiba, [2017]. Disponível em: https://docs.ufpr.br/~marcosantonio/disciplinas/algIII/sequencial_indexado.pdf. Acesso em: 26 set. 2020.

SOMMERVILLE, Ian. **Engenharia de Software**. Engenharia de Software, 2003. disponível em: <http://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>. Acesso em: 5 de Out. 2020

SOPRANA, Paula. **Corononavírus cria corrida por programadores de COBOL, linguagem dos anos 1960**. São Paulo, [2020]. Disponível em: <https://www1.folha.uol.com.br/tec/2020/04/ibm-convoca-programadores-de-cobol-linguagem-dos-anos-1960-durante-pandemia.shtml>. Acesso 26 set. 2020.

TURBAN, Efraim; VOLONINO, Linda. **Tecnologia da Informação para gestão**. Porto Alegre: Bookman, 2013.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Supervisor(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a): Luiz Carlos Burigo

Avaliador(a): Everaldo Artur Grahl

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?		X	
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		

PARECER – PROFESSOR AVALIADOR: (PREENCHER APENAS NO PROJETO)

O projeto de TCC ser deverá ser revisado, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 5 (cinco) tiverem resposta ATENDE PARCIALMENTE.

PARECER: (X) APROVADO () REPROVADO

Assinatura: Everaldo Artur Grahl Data: 06/12/2020

Está grifado em amarelo o que melhorar – seja texto, formato ou item já falado na avaliação anterior – por exemplo diagramas da UML. No mais tudo OK.

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.