

CURSO DE SISTEMAS DE INFORMAÇÃO – TCC ACADÊMICO		
(<input checked="" type="checkbox"/>) PRÉ-PROJETO	(<input type="checkbox"/>) 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).

Um dos diferenciais que leva o Cobol a ter maior performance e custo-benefício é a utilização de arquivos para armazenamento de dados cuja organização 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 deve 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 e também 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 ferramentas. Partindo da necessidade de disponibilizar acesso aos dados de uma aplicação Cobol, como é possível realizar este feito em *real time* com o sistema legado em pleno funcionamento? **E por** fim, como é possível integrar tecnologias que estejam disponíveis para modelagem desses dados?

1.1 OBJETIVOS

O objetivo geral do projeto é disponibilizar um protótipo para a integração de dados mantidos por uma aplicação Cobol a um novo modelo de banco de dados SGDB NoSQL na *cloud*. Os objetivos específicos são:

- a) modelar arquivo Cobol que receberá os dados a serem migrados;

- b) modelar a arquitetura a ser implementada para migração de dados;
- c) implementar programas Cobol que farão a extração de dados;
- d) definir e modelar as novas coleções na *cloud* em um banco de dados MongoDB;
- e) criar scripts responsáveis por enviar os dados extraídos para um sincronizador de dados com o banco na *cloud*;
- f) eleger uma tecnologia para integração com a nova base de dados na *cloud* para modelar analiticamente os dados.

2 TRABALHOS CORRELATOS

Foram observados três trabalhos correlatos que apresentam propostas semelhantes ao projeto proposto. Andre Luiz Jacinto (2006), propôs uma ferramenta de extração de dados de um banco de dados restritos a aplicação COBOL através de comandos de SQL. Fábio Ivan Borchardt (2016), construiu uma API que busca a integração entre o sistema legado e novas tecnologias. Rodrigo Gonçalves et al. (2019) que buscou a extração e migração de dados para um novo banco de dados não relacional.

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

A ferramenta construída por Jacinto (2006) possibilita ao usuário a utilização de comandos DDL (Data Definition Language). Os comandos são enviados a um interpretador que faz uma análise léxica, sintática e semântica dos comandos e 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.

Na escolha de tecnologias o autor utilizou-se de Java, a estratégia utilizada foi a implementação de um motor de templates que é instanciado por uma requisição com os comandos de leitura, a aplicação associa os valores das variáveis aos blocos de templates utilizados de acordo com as palavras reservadas utilizadas (JACINTO, 2006).

2.2 MELHORIA DE SISTEMA LEGADO DO SETOR AGROPECUARIO UTILIZANDO API

Para modernizar o sistema agropecuário da empresa Chip Inside foi proposto a criação de uma API que disponibilize informações importantes para tomada de decisão da fazenda

através de integração de outras tecnologias como aplicações web ou mobile que consumem dados da API proposta. 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 simples comando identificar toda a estrutura do banco utilizado (com seus atributos e relacionamentos) do sistema legado e salvar as mesmas informações em uma classe denominada Model (BORCHARTDT,2016).

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

A empresa onde os autores implementaram a aplicação estava tendo problemas com o volume excessivo de dados ao qual o seu sistema operava, com cada vez mais usuários e dados a serem salvos a estrutura de consistência de dados do banco relacional utilizado começou a apresentar problemas de desempenho, o tempo de execução e de consultas começou a ficar lento (MACHADO et al., 2016).

Para mitigar o problema os autores trazem como proposta de migração de dados do banco legado para uma nova estrutura em MongoDB, um banco NoSQL que no contexto do problema apresentado teria apenas função para leitura. Os autores implementaram a solução de maneira que os dados do banco legado um DB2 conforme *query* previamente definida migraria dados para o banco de dados MongoDB (MACHADO et al., 2016).

A migração pode acontecer de duas maneiras sendo uma na forma de substituição onde a API apaga todos os dados existentes no banco MongoDB e realiza novamente toda migração de dados e a outra forma seria em um modelo automático onde a API apenas atualizaria o banco em MongoDB a cada atualização no banco DB2 mantendo assim o banco DB2 com a função de escrita e o banco MongoDB com função de leitura (MACHADO et al., 2016).

3 PROPOSTA DO PROTÓTIPO

Na subseção 3.1 deste capítulo serão apresentadas as justificativas para realização do trabalho proposto. Serão descritos os principais requisitos do protótipo na subseção 3.2, a metodologia e o cronograma planejado para o desenvolvimento do trabalho na subseção 3.3.

3.1 JUSTIFICATIVA

Em abril de 2020, segundo reportagem da Folha de São Paulo (SOPRANA, 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 um sistema legado tem para com um sistema financeiro governamental americano mesmo nos dias atuais, assim como 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 sistemas legados e seus dados a novas tecnologias, mas 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 a dados mantidos por sistemas legados assim como apresentar uma melhoria no sistema legado.

Quadro 1 –Comparativo entre os trabalhos correlatos

Correlatos Características	Luiz Jacinto (2006)	Fábio Ivan Borchardt (2016)	Machado et al. (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 toda uma estrutura que possibilite 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 e também manter uma estrutura performática de extração de dados visando a necessidade de consultar grandes volumes de dados.

A característica de integração de sistemas legados a novas tecnologias é presente em Borchardt (2016), sua API é uma web service consumida 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 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 a dividir o banco de dados para um banco 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 real time conforme query previamente definida para fazer o *insert*, *update* e *delete* em *real time* na base de leitura.

Com base nas características apresentadas no quadro 1, é possível identificar muitas semelhanças entre os trabalhos correlatos e o trabalho aqui proposto. A proposta tem como objetivo migrar todos os dados de um banco de dados restrito a aplicação Cobol para um SGDB modelando de forma a ter uma base de dados performática e possibilitando efetuar consultas de dados de forma flexível, assim como 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 apresentar uma arquitetura de integração que:

- a) permita 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;
- 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*) (RF);
- i) o programa de sincronização deverá gravar as informações recebidas em um cluster de MongoDB na *cloud*.

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 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 persiste os registros de todos os clientes.

Existem duas maneiras de acessar arquivos restritos a 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 quando física (SCHREINER, 2017).

Podemos organizar os arquivos 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 temos um grande volume de dados armazenados este método de armazenamento sequencial e acessos aleatórios podem deixar o sistema lento,

sendo indicado para o armazenamento de grandes volumes de dados onde é necessário um acesso direto o armazenamento em arquivos sequenciais indexados (SCHREINER, 2017).

Marco Antonio Schreiner 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 sendo que cada índice referência um bloco de dados e a área de excedentes reservada para o acréscimo de novos registros que não podem ser colocados na área principal (SCHREINER, 2017). Neste sentido Marco Antonio Schreiner cita como características dos arquivos sequenciais indexados um acesso aleatório satisfatório, facilidade de busca, inserções e exclusões.

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).

REFERÊNCIAS

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

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

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.

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.

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.

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.

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.

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

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.

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

Chaves, Luciano L. **Sistemas Legados e a Aplicação de Processo de Reengenharia de Software**. São Paulo, [2004]. Disponível em: https://www.researchgate.net/profile/Luciano_Chaves/publication/261623676_Sistemas_Legados_e_a_Aplicacao_de_Processos_de_Reengenharia_de_Software/links/00b49535e9a2f86b45000000/Sistemas-Legados-e-a-Aplicacao-de-Processos-de-Reengenharia-de-Software.pdf. Acesso em: 26 set. 2020.

Bispa, 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.

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

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		
	1. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?		X	
	2. DESCRIÇÃO DO SISTEMA ATUAL O sistema atual está claramente descrito e embasa de modo consistente o sistema proposto?	X		
	4. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	5. JUSTIFICATIVA O sistema proposto está descrito de forma adequada e abrange soluções para os problemas do sistema atual?	X		
	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		
	6. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?		X	
	7. 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		
	8. 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	9. 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: 04/11/2020

OBS: Poucos ajustes a fazer no texto. Está grifado onde melhorar. Deixar mais claro objetivos específicos e talvez excluir alguns. Nos requisitos faltou classificar alguns. Na metodologia dizer quais diagramas da UML serão adotados.

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.