



UNIVERSIDADE DO VALE DO TAQUARI - UNIVATES

CURSO DE SISTEMAS DE INFORMAÇÃO

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA AUXILIAR NA
MIGRAÇÃO DE DADOS ENTRE SISTEMAS ERP**

Leandro Vogel

Lajeado, dezembro de 2019

Leandro Vogel

DESENVOLVIMENTO DE UMA FERRAMENTA PARA AUXILIAR NA MIGRAÇÃO DE DADOS ENTRE SISTEMAS ERP

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas de Informação da Universidade do Vale do Taquari - Univates, como parte dos requisitos para a obtenção do título de bacharel em Sistemas de Informação.

Orientador: Prof. Me. Fabrício Pretto

Lajeado, dezembro de 2019

AGRADECIMENTOS

Primeiramente agradeço a minha família, por todo apoio, paciência e compreensão em todos esses anos cursando a faculdade, em especial a minha mãe Dulce por ser o alicerce desta longa jornada, sem o seu apoio este sonho não seria possível.

Agradeço a minha esposa Alessandra pela compreensão e carinho nos momentos difíceis e pelo incentivo que me deu forças na reta final para que conseguisse superar este desafio.

Ao meu orientador, Prof. Me. Fabrício Pretto pelo companheirismo e apoio em todo o desenvolvimento do trabalho.

Finalmente, agradeço ao meu filho Luís Antônio, que nasceu nestes últimos meses e que foi o incentivo que faltava nesta reta final de curso.

RESUMO

Com as constantes mudanças fiscais e tecnológicas, muitos sistemas ERP (*Enterprise Resource Planning*) acabam se tornando obsoletos não atendendo às necessidades das empresas, provocando a troca do sistema e consequente migração das informações presentes no banco de dados. Este trabalho acadêmico teve como objetivo desenvolver uma ferramenta para auxílio no processo de migração de banco de dados entre sistemas ERP. Como benefícios da ferramenta desenvolvida, destacam-se agilidade e redução do custo. Para tanto, na fundamentação teórica foram abordados conceitos de autores das áreas de sistemas de informação, banco de dados, migração de dados, entre outros. Como metodologia, utilizou-se pesquisa bibliográfica e experimental. A ferramenta automatiza a criação de conexões entre os bancos de dados, avaliação de tabelas e campos do banco de origem, visualização dos dados e a realização de vínculos e criação de scripts para a migração dos dados. Como resultados, são apresentados o relato da implementação e os ganhos obtidos com a utilização desta ferramenta no processo de migração entre bancos de dados.

Palavras-chave: Banco de Dados, ERP, Migração de Dados.

ABSTRACT

With the constant fiscal and technological changes, many systems, ERP (Enterprise Resource Planning) end up becoming obsolete not meeting the needs of enterprises, causing the system change and consequent migration of information existing in the database. This academic work aimed to develop a tool to aid in the process of database migration between ERP systems. As benefits of the developed tool, agility and cost reduction stand out. Therefore, in the theoretical basis, concepts of authors from the areas of information systems, database, data migration, among others were addressed. As a methodology, bibliographic and experimental research was used. The tool automates the creation of connections between databases, assessment of tables and fields in the source database, data visualization and realization of bonds and creation of scripts for data migration. As results, the implementation report and the gains from using this tool in the database migration process are presented.

Palavras-chave: Data Base, ERP, Data Migration

LISTA DE FIGURAS

Figura 1 - Esquema de um sistema de informação	18
Figura 2 - Estrutura ERP	21
Figura 3 - Modelo em cascata ou ciclo de vida do software.....	24
Figura 4 - Ciclo de vida de sistemas ERP	25
Figura 5 - Conceitos de Banco de Dados.....	27
Figura 6 - Estrutura do Banco de Dados.	27
Figura 7 - Representação de registro.....	28
Figura 8 - Tipos e atributos.	29
Figura 9 - Estratégia de migração Butterfley.....	33
Figura 10 - Modelo entidade-relacionamento	45
Figura 11 - Exemplo código fonte tipos de banco	47
Figura 12 - Exemplo código fonte leitura estrutura banco	48
Figura 13 - Tela Inicial.....	49
Figura 14 - Tela Manter Projetos.....	50
Figura 15 - Tela de Conexões	51
Figura 16 - Tela de <i>Scripts</i>	53
Figura 17 - Seleção do tipo de <i>script</i>	55
Figura 18 - Tela de Migração: primeira e segunda parte	56
Figura 19 - Tela de Migração: terceira parte	57
Figura 20 - Tela de Migração: exemplo do painel de configurações do tipo de <i>script create</i>	59

Figura 21 - Tela de Migração: exemplo do painel de configurações do tipo de <i>script delete</i>	60
Figura 22 - Tela de Migração: exemplo do painel de configurações do tipo de <i>script update</i>	61
Figura 23 - Tela de Migração: exemplo do painel de configurações do tipo de <i>script insert</i>	62
Figura 24 - Seleção do tipo de relacionamento	64
Figura 25 - Tela de Relacionamento	65
Figura 26 - Tela de Dados	66
Figura 27 - Código fonte exemplo de separação entre teste e migração final	67
Figura 28 - Código fonte exemplo tipo <i>script insert</i>	68

LISTA DE QUADROS

Quadro 1 - Ciclo de vida do software ERP x ciclo de vida do software tradicional	26
Quadro 2 - Estratégias de migração de dados	31
Quadro 3 - Prioridades dos requisitos	43
Quadro 4 - Requisitos funcionais	43
Quadro 5 - Requisitos não funcionais	44
Quadro 6 - Detalhamento das tabelas do banco de dados	45
Quadro 7 - Análise da ferramenta.....	71

LISTA DE ABREVIATURAS

ER	Entidade-Relacionamento
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
IDE	Interface Development Environment
SGDB	Sistema de Gerenciamento de Banco de dados
SQL	Structured Query Language
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Definição do problema	13
1.2	Delimitação do estudo	14
1.3	Objetivos	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
1.4	Justificativa	15
1.5	Estrutura do trabalho	16
2	REFERENCIAL TEÓRICO	17
2.1	Sistemas de informação	17
2.1.1	Informação	19
2.1.2	Sistemas integrados	20
2.1.3	Enterprise Resource Planning (ERP)	20
2.1.4	Implantação de sistemas ERP	23
2.1.4.1	Ciclo de vida do software	24
2.2	Banco de dados	26
2.2.1	Estrutura do banco de dados	27
2.2.1.1	Tabela / Relação	27
2.2.1.2	Linhas / Tuplas	28
2.2.1.3	Coluna / Atributo	28
2.2.2	Dados	29
2.3	Migração de dados	30
2.3.1	Estratégias	30
2.3.2	Metodologias	31
2.3.2.1	Chicken Little	31
2.3.2.2	Butterfley	32
2.3.2.3	ETL	33
3	METODOLOGIA	34
3.1	Caracterização da pesquisa	34
3.1.1	Pesquisa bibliográfica	34

3.1.2	Pesquisa exploratória	34
3.1.3	Pesquisa quantitativa	35
3.1.4	Pesquisa experimental	35
3.2	Ferramentas	36
3.2.1	Linguagem de programação Java	36
3.2.2	Netbeans 8.2.....	36
3.2.3	Banco de dados PostgreSQL.....	37
4	TRABALHOS RELACIONADOS.....	38
4.1	Análise comparativa das ferramentas de ETL – Kettle e Talend.....	38
4.2	Processos de implantação e migração de dados com utilização de ETL para um ERP comercial	39
4.3	Requisitos para ferramentas de migração de dados.....	39
4.4	Migração e testes de um ERP do SGDB SQL Server para o SGDB PostgreSQL.....	40
5	IMPLEMENTAÇÃO DA FERRAMENTA	41
5.1	Cenário avaliados	42
5.2	Modelagem da ferramenta	42
5.2.1	Levantamento de requisitos	43
5.2.1.1	Requisitos funcionais	43
5.2.1.2	Requisitos não funcionais.....	44
5.2.2	Modelo entidade-relacionamento	45
5.3	Desenvolvimento da ferramenta	46
5.3.1	Classes e estrutura de dados	46
5.3.2	Telas e funcionalidades da ferramenta	48
5.3.3	Rotina de exportação e importação de dados.....	67
5.3.4	Dificuldades encontradas	69
6	ANÁLISE DA FERRAMENTA	71
7	CONSIDERAÇÕES FINAIS.....	74
8	TRABALHOS FUTUROS	76
	REFERÊNCIAS.....	78

1 INTRODUÇÃO

Atualmente a grande maioria dos estabelecimentos comerciais ou entidades, utilizam sistemas ERP (*Enterprise Resource Planning*) para o gerenciamento e organização das suas atividades. Nas últimas décadas, houveram novas exigências por parte do governo federal e estadual, como por exemplo, nota fiscal eletrônica e nota fiscal consumidor, que obrigou adequações dos sistemas de gestão. Com isso muitos sistemas tornaram-se obsoletos, por não conseguirem acompanhar as constantes mudanças.

Um sistema ERP segundo Caiçara (2015), é um sistema de informação que permite a integração de dados entre sistemas de informação tradicionais como também processos de negócios de uma organização. Diante dessa definição, várias empresas vêm procurando não somente uma ferramenta para atender as demandas de seus clientes e obrigações legais, mas também para ajudar no planejamento estratégico, auxiliando na tomada de decisões. A velocidade com que estas demandas chegam e se transformam, torna muitos sistemas ultrapassados, obrigando empresa a trocar de sistema gestão.

O banco de dados é uma peça fundamental para os sistemas ERP, ele torna os processos operacionais e de busca da informação, mais ágeis e eficientes. No banco de dados são armazenadas as informações geradas pelo sistema, e é através dessas, que são apurados os resultados estratégicos para a melhor tomada de decisão da empresa. “Banco de dados é uma coleção de dados que, tipicamente, descreve as atividades de uma ou mais organizações relacionadas” (RAMARKRISHNAN, 2012, p. 3).

O processo de implantação de sistema é dividido em várias etapas, e, segundo MENDONÇA (2009, p. 22), “Uma etapa fundamental na troca de um sistema ERP é a migração de dados. O processo de transferir dados de um sistema para outro é chamado de migração de dados”. Esse processo é considerado um dos mais importantes, tendo em vista o valor que as informações contidas no banco de dados têm para a empresa. A perda dessas informações pode ocasionar prejuízos financeiros a empresa.

Pensando nesse cenário de troca de sistemas ERP e a importância da integridade e segurança das informações, apoiados pela dificuldade e responsabilidade no processo de migração de dados, o presente trabalho apresenta o desenvolvimento de uma ferramenta que auxilia empresas de tecnologia no processo de troca de um sistema ERP, tornando o processo ágil, eficaz e eficiente, diminuindo o tempo de migração e o risco de erro na troca de informações entre as bases de dados.

Diante destas questões, a ferramenta proposta tem como finalidade melhorar e agilizar o processo de migração do banco de dados, e, com isso, aumentar a satisfação do cliente com o serviço prestado bem como reduzir custo de implantação.

Nos tópicos seguintes, são apontados: a delimitação do estudo, o objetivo geral, os objetivos específicos, e ao final do capítulo, é apresentada a estrutura do trabalho.

1.1 Definição do problema

A agilidade nos processos operacionais e o planejamento estratégico têm se tornado cada vez mais importante para as organizações, sendo consideradas variáveis relevantes para os resultados desejados. Diante disto, muitas organizações vêm aumentando sua exigência com seus sistemas de gestão, já que por meio deste é possível gerenciar as informações, prever os materiais necessários para atender as demandas imediatas, planejar a maneira mais adequada de suprir as necessidades futuras e definir estratégias para melhorar o desempenho de suas atividades.

No que tange aos sistemas de informação, cada instituição precisa planejar e organizar os dados inseridos no seu sistema ERP de maneira que consiga atender as necessidades de seus usuários, otimizando a forma de gerir seus processos e melhorar o aproveitamento de seus recursos. Em se tratando especificamente da

etapa de migração de banco de dados, a qual pode ser considerada mais complexa que outras na implantação de sistemas, deve-se ter um controle mais apurado dos processos, visto que muitos dos itens que compõem o procedimento são de alta importância para a empresa que está realizando a troca do seu software de gestão. Desta maneira, pode-se considerar que a troca de sistema ERP de uma empresa é composta de várias etapas, dentre elas: levantamento da necessidade do cliente, a avaliação do banco de dados atual, da infraestrutura, custo da implantação, migração de dados, dentre outros. Empresas conceituadas como SAP e Oracle desenvolveram ferramentas para auxiliar na migração de seus produtos, um exemplo disto é o SAP Data Migration, que permite uma migração de dados *end-to-end*.

Ao realizar a implantação de um sistema ERP, é necessário evitar desperdícios dos recursos financeiros disponibilizados para a atividade, reduzindo problemas com a troca de sistema, evitar a perda e assegurar o armazenamento adequado das informações, sendo possível otimizar ao máximo todas as etapas e com isso aumentar a qualidade dos serviços prestados.

A migração eficiente de dados também é importante para minimizar situações de falta e perda de informação fundamentais para o trabalho da empresa, fato que pode comprometer o desempenho da empresa, sendo assim, o presente estudo busca desenvolver uma ferramenta para auxiliar na migração de dados entre sistemas ERP, visto que atualmente algumas empresas de tecnologia ainda realizam o processo totalmente manual sem utilizar nenhuma. Dentre os principais problemas pode-se mencionar o tempo de migração, a falta de integridade das informações, falta de mão de obra especializada e o tempo escasso para migração. Diante do exposto, levanta-se a seguinte questão de pesquisa: Como o desenvolvimento de uma ferramenta auxiliará no processo de migração de bases de dados?

1.2 Delimitação do estudo

O estudo tem seu escopo compreendido em descrever o processo de migração de banco de dados de empresas de tecnologia da informação que fornecem sistemas ERP de pequeno e médio porte. A proposta é desenvolver uma ferramenta para auxiliar migrações dos bancos de dados relacionais que geralmente são utilizados por estas empresas. Os gerenciadores de bancos utilizados para este estudo serão PostgreSQL, Firebird e SQL Server.

A ferramenta não será responsável por validação de dados pós migração, esse processo consiste em conferir os dados no sistema ERP antigo e confrontar com o novo. Essa validação será feita manualmente pelos responsáveis do projeto de migração.

1.3 Objetivos

Os objetivos são divididos em objetivo geral e objetivos específicos.

1.3.1 Objetivo Geral

Desenvolver uma ferramenta para auxiliar na migração de banco de dados entre sistemas ERPs, visando tornar o processo ágil, eficiente e eficaz, diminuindo erros de importação e o tempo de implantação do novo software.

1.3.2 Objetivos Específicos

- Elaborar o projeto de desenvolvimento da ferramenta;
- Mensurar o tempo e custo demandado nas migrações realizadas manualmente antes do desenvolvimento da ferramenta;
- Mensurar o tempo e custo no processo de migração de banco de dados utilizando a ferramenta desenvolvida;
- Avaliar a eficiência e eficácia da ferramenta desenvolvida confrontando os resultados do processo atual de migração de dados com os resultados obtidos com a ferramenta desenvolvida.

1.4 Justificativa

A procura pela melhoria dos processos dentro das organizações e mudanças de legislação é constante. Diante disso é preciso desenvolver novos métodos para o desempenho das atividades, utilizando racionalmente os recursos disponíveis. Perante este novo posicionamento organizacional, é necessário que as organizações tenham um sistema de gestão capaz de promover a melhoria no fluxo das informações e maior agilidade no desenvolvimento das atividades e com isso as empresas vêm exigindo mais dos seus sistemas ERP.

Com o sistema não conseguindo mais atender as demandas da empresa, esta se vê obrigada a procurar novas soluções, novos sistemas, e surge a necessidade da troca da ferramenta atual, com isso é necessário fazer a migração dos dados entre o sistema velho e o novo. O processo de migração de dados realizado por empresas de tecnologia geralmente é oneroso, e isso muitas vezes inviabiliza a troca do sistema.

1.5 Estrutura do trabalho

O presente trabalho está dividido em oito capítulos. O primeiro capítulo consiste na introdução do trabalho, que tem como objetivo nortear sobre o assunto que será abordado. O capítulo dois apresenta os referenciais teóricos que embasam os conceitos relacionados às terminologias e tecnologias que serão utilizadas. A metodologia e técnica de pesquisa é exposta no capítulo três, seguida trabalhos relacionados que ocorre no capítulo quatro. O capítulo cinco foi reservado para escrita da implementação de como a ferramenta foi desenvolvida, seguida pela sua análise descrita no capítulo seis. Por fim o trabalho encerrou com as considerações finais que o autor teve em cima do trabalho no capítulo sete e trabalhos futuros que são melhorias que a ferramenta no futuro.

2 REFERENCIAL TEÓRICO

Neste capítulo é apresentado o embasamento teórico para o estudo dos principais conceitos relacionados à sistemas de informação, banco de dados e migração de dados.

2.1 Sistemas de informação

Conforme Laudon (2004, p. 7) sistema de informação é “tecnicamente como um conjunto de componentes inter-relacionados que coleta (ou recupera), processa, armazena e distribui informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização”.

De acordo com Rosini e Palmisano (2012, p. 3) “o conceito básico de sistemas de informações estabelece que todo sistema é um conjunto de elementos interdependentes em interação, visando atingir um objetivo comum”. Os autores complementam afirmando que existem dois tipos de sistemas: o aberto, que recebe influência do meio ambiente e o influencia, e o fechado, que não sofre influência, mas também não o altera.

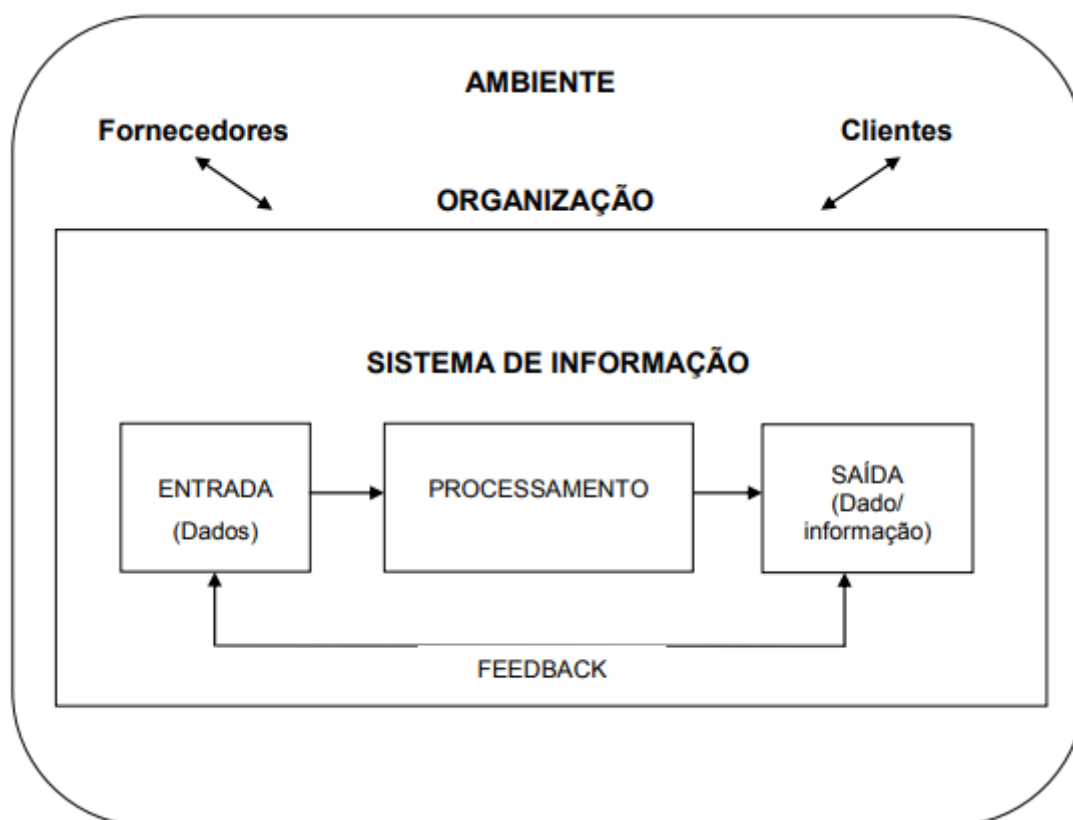
Os sistemas de informação não podem ser ignorados pelas organizações, pois atualmente desempenham um papel importante no desenvolvimento das estratégias empresariais. Conforme Belmiro (2012), eles auxiliam no melhoramento da eficiência das operações, na criação de novos produtos ou serviços, estreitam o relacionamento com clientes e fornecedores, ajudam na tomada de decisão e garantem uma vantagem competitiva para as organizações.

Ainda segundo Belmiro (2012), existem três operações básicas em um sistema de informação que trazem resultado para as organizações, que são:

- Entrada: que consiste na coleta de dados brutos de dentro e fora da empresa;
- Processamento: transforma os dados em informações;
- Saída: transfere as informações processadas aos responsáveis pela sua utilização na empresa.

Após passar por esse processo, o sistema de informação precisa de um *feedback*, que é a entrada e volta para os membros da organização responsáveis por avaliar e/ou corrigir a etapa de entrada, possibilitando a auto regulação do sistema, representada na Figura 1.

Figura 1 - Esquema de um sistema de informação



Fonte: elaborado pelo autor, adaptado de Laudon e Laudon (2007, p. 10).

2.1.1 Informação

A informação, segundo Audy, Andrade e Cidral (2007, p. 94), “é uma coleção de fatos organizados de forma a possuir um valor adicional aos fatos em si”. Ainda conforme os autores, são dados relacionados que já passaram por um processo de transformação, sendo seu conteúdo adequado para uso específico.

Silva (2010), afirma que se as informações forem utilizadas de maneira correta, elas podem mudar a forma de atuação da organização, assim como seu relacionamento com outras empresas. Ainda segundo Silva (2010, p. 301) “a informação é o dado útil para a tomada de decisão e a solução de problemas”. Diante dos conceitos anteriores, podemos definir que a informação é um conjunto de dados organizados de forma a terem significado, valor e finalidade para o seu destinatário. Uma informação consiste em um cruzamento de dados.

Existem momentos em que a informação é colocada em risco e esses momentos compõem e identificam o ciclo de vida da informação (SÊMOLA, 2003). São eles:

- Manuseio: quando a informação é criada ou alterada;
- Armazenamento: quando a informação é retida (por exemplo, em meio magnético ou banco de dados);
- Transporte: quando a informação é conduzida ou transportada (por exemplo, por fax, e-mail ou migração de dados);
- Descarte: quando a informação é inutilizada, descartada (por exemplo, quando um registro, arquivo eletrônico ou papel é excluído).

Em ambientes corporativos, é muito grande a quantidade de informações e constante a manipulação de dados. O armazenamento de destas informações deve ser estudado e aplicado, pois qualquer descuido pode colocá-las em risco, deixando-as suscetíveis a perdas e roubos GALVÃO (2015).

Conforme Galvão (2015), a segurança da informação é caracterizada e assegurada levando em consideração os seguintes princípios:

- Confidencialidade: a confidencialidade consiste, em garantir que somente pessoas autorizadas tenham acesso ao conteúdo das informações;

- **Integridade:** o princípio da integridade se baseia na garantia de que as informações armazenadas estão corretas, são verdadeiras e não sofreram nenhum tipo de violação, isto é, que qualquer alteração, redução ou acréscimo foram autorizados por seus proprietários;
- **Disponibilidade:** a disponibilidade é a garantia de que as informações estarão disponíveis sempre que for solicitada pelas pessoas autorizadas;
- **Autenticidade:** a autenticidade é a garantia da legitimidade da transação, do acesso, da comunicação e da própria informação, isto é, assegura a veracidade da fonte anunciada e de que a informação não sofreu nenhum tipo de modificação não autorizada ao longo do processo;
- **Irrevogabilidade:** a irrevogabilidade garante que uma informação seja irrevogável, ou inegável, isto é o autor de uma transação feita anteriormente não seja capaz de negar tal autoria sobre o ato.

2.1.2 Sistemas integrados

De acordo com Laudon e Laudon (2007), os sistemas integrados oferecem um conjunto de módulos de software integrado e um banco de dados central, que permite que os dados sejam compartilhados entre as diferentes áreas da organização.

Conforme Belmiro (2012), os aplicativos integrados são sistemas que abrangem todas as áreas da empresa, desde níveis operacionais até a gerência. Eles auxiliam as empresas a tornar seus processos mais flexíveis, pois é como se vários tipos de sistema estivessem trabalhando em conjunto.

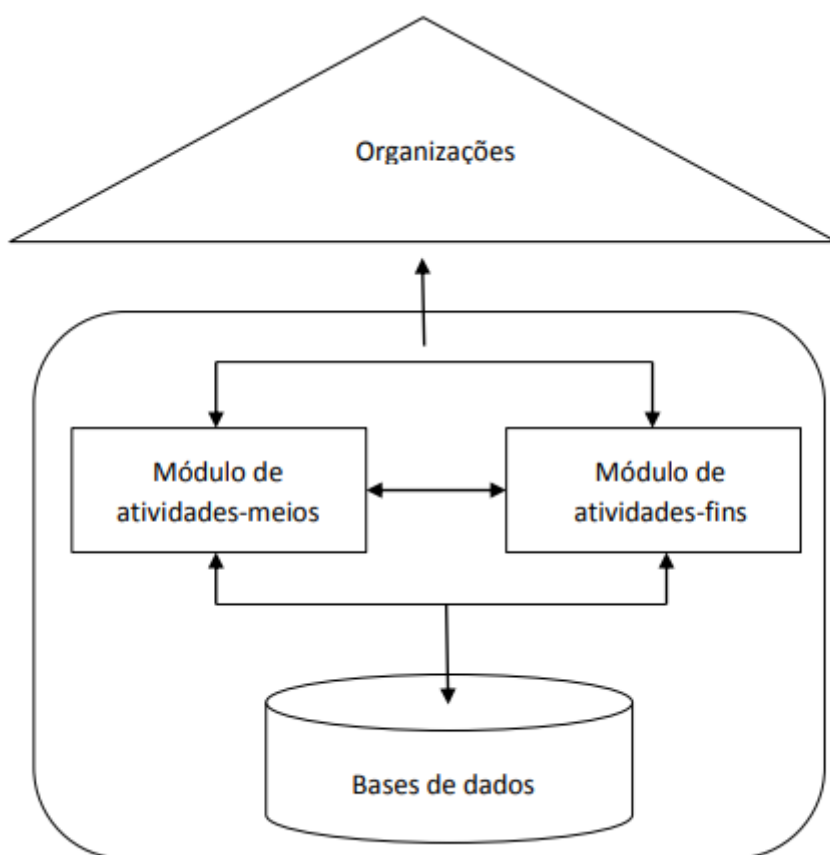
2.1.3 Enterprise Resource Planning (ERP)

De acordo Santos (2003), ERP é um conjunto de processos instantâneos que atualizam outros processos de forma automática. Já para Haberkon (2009) ERP é um sistema que consiste em conciliar a parte financeira, contábil, controle dos ativos, RH e gestão de conhecimento. De acordo com Caiçara (2015, p. 96).

Podemos definir didaticamente ERP como um sistema de informação adquirido na forma de pacotes comerciais de software que permitem a integração entre dados dos sistemas de informação Transacionais e dos processos de negócios de uma organização.

Haberkon (2009) define a estrutura do ERP como um grande guarda-chuva com dois apêndices, um como atividade fim e o outro com as atividades meio. Normalmente o ERP possui dois conjuntos básicos de módulos, representados na Figura 2, um está direcionado a atividade fim ou atividade vertical da organização, o outro conjunto dá suporte a atividades meios que tem como principal objetivo cuidar da parte administrativa da empresa (SANTOS, 2003).

Figura 2 - Estrutura ERP



Fonte: elaborado pelo autor, adaptado de Santos (2003, p. 171).

Alguns pontos e características importantes dos sistemas ERP devem ser cuidadosamente analisados conforme Gomes e Vanalle (2001):

- Os ERPs são pacotes comerciais desenvolvidos a partir de modelos padrões de processos. Devido à grande experiência da empresa desenvolvedora do software em diferentes ramos de negócios, são desenvolvidos padrões de processos, os quais não são específicos para

uma determinada necessidade, e sim genéricos podendo a empresa adequar-se ou não a eles;

- Integram todas as áreas da empresa, sendo este um grande ganho na utilização destas ferramentas. A empresa obtém integridade e confiabilidade nas informações adquiridas através do sistema, pois a entrada de um dado ocorre uma única vez dentro do sistema, que a partir de então passa a atualizar automaticamente todos os módulos necessários;
- Permitem a adequação das funcionalidades existentes no sistema da empresa através do processo de parametrização. Este processo consiste na definição de diversos valores que são introduzidos no sistema com o intuito de dimensionar o perfil da empresa e o comportamento do sistema;
- Possibilitam o processo de customização de determinados processos que não se adequam de maneira nenhuma à empresa, mesmo fazendo uso da parametrização. A customização é a adaptação do sistema às necessidades específicas da empresa, onde é necessário intervir com programas ou rotinas que se integram ao ERP;
- Possuem custos elevados, entre eles destacam-se os custos de hardware e infraestrutura computacional, de aquisição da licença de uso do ERP e treinamento e consultoria para a implantação. Um sistema de ERP apresenta muitas complexidades, sendo que sua implantação deverá ser realizada por profissionais que conheçam não somente o negócio da empresa, como também a solução escolhida;
- Os sistemas desenvolvidos em outros países, adaptam-se à realidade brasileira no que diz respeito às obrigações legais, através do processo chamado de localização;
- Os fornecedores liberam periodicamente versões atualizadas (*upgrades*) que agregam melhorias, correções de problemas e erros do sistema. Este processo de atualização deve ser flexível e permitir a adequação da nova versão com possíveis customizações efetuadas no produto;

- Forçam na maioria das vezes alterações nos processos produtivos e administrativos. Será necessário tanto a adaptação do sistema aos processos da empresa, como a adaptação da empresa a determinados processos do sistema;
- Impactam sobre os recursos humanos da empresa, pois as pessoas terão que se preocupar com o processo como um todo e não apenas com a sua atividade específica. Devido a integração do sistema, um problema de uma área poderá se alastrar rapidamente para outros departamentos, existindo o risco de chegar a afetar toda a empresa;
- Apresentam dificuldade no cumprimento de prazos e orçamentos, causados por: resistência por parte das pessoas, rotatividade dos funcionários que foram treinados no novo sistema ou que dominam o negócio da empresa, qualidade dos recursos humanos internos e da equipe de consultoria contratada, limitações inerentes ao próprio produto ERP escolhido, dificuldade de integrar o ERP com outros sistemas existentes dentro da empresa ou corporação.

2.1.4 Implantação de sistemas ERP

De acordo com Sommerville (2011), a implantação é o processo de concretização do projeto de software em um programa. Ainda segundo o autor, a fase final do processo de desenvolvimento, é quando o software é instalado no hardware e está preparado para operar. Normalmente isso pode envolver mais configurações de sistema, a transferência de dados dos sistemas existentes, bem como a preparação da documentação e o treinamento de usuários. Nessa fase, pode ser preciso reconfigurar os outros sistemas para assegurar de que o novo sistema interopera com eles. Conforme os autores Stair e Reynolds (2015), atualmente muitas organizações estão contratando empresas externas para executar seu projeto e as funções de implantação, manutenção e revisão de sistemas.

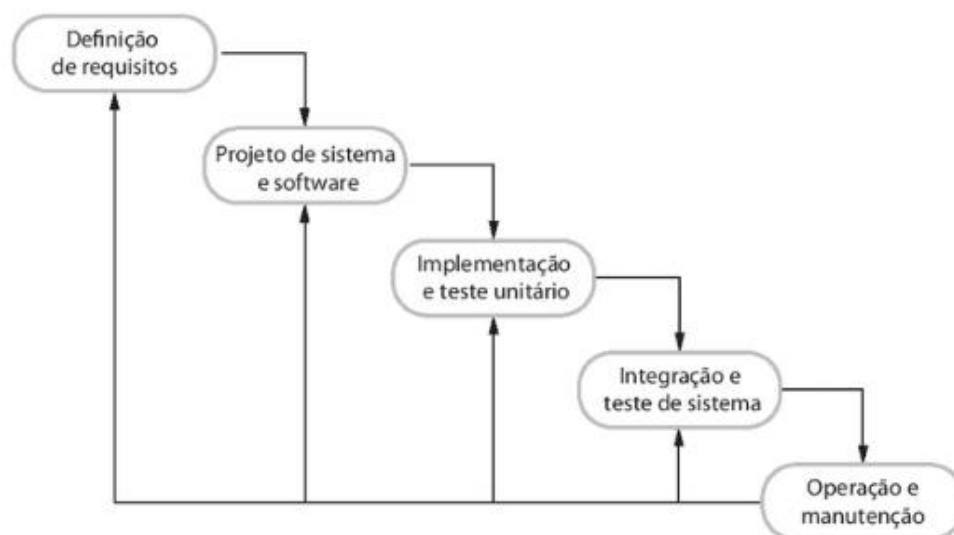
Embora a implantação de sistemas seja uma fase simples, em princípio, muitas dificuldades podem ocorrer durante esse processo. O ambiente de usuário pode ser diferente daquele testado pelos desenvolvedores do sistema e adaptar o software para lidar com diversos ambientes de usuário pode ser complicado. Os dados

existentes podem exigir limpezas extensivas e partes deles podem estar faltando. As interfaces com outros sistemas podem não estar devidamente documentadas (SOMMERVILLE, 2011).

2.1.4.1 Ciclo de vida do software

Embora a implantação de sistemas seja uma fase simples, em princípio, muitas dificuldades podem ocorrer durante esse ciclo. O ambiente de usuário pode ser diferente daquele testado pelos desenvolvedores do sistema e adaptar o software para lidar com diversos ambientes de usuário pode ser complicado. Os dados existentes podem exigir limpezas extensivas e partes deles podem estar faltando. As interfaces com outros sistemas podem não estar devidamente documentadas (SOMMERVILLE, 2011). A Figura 3 demonstra este ciclo de vida do software.

Figura 3 - Modelo em cascata ou ciclo de vida do software

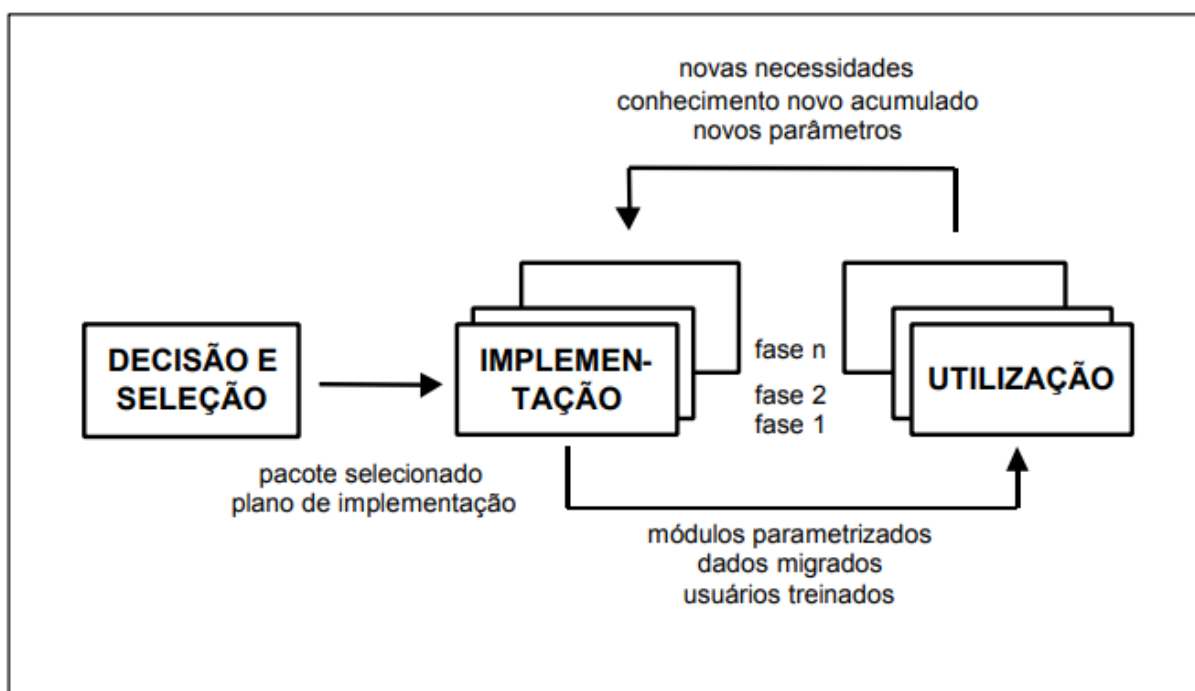


Fonte: Sommerville (2011, p. 20).

Conforme Souza e Zwicke (2000), há diferenças entre o ciclo de vida do software de pacotes comercial e o ciclo de vida do software tradicional, de acordo com a citação e representação na Figura 4.

O ciclo de vida de pacotes comerciais deve ser considerado de maneira diferente dos modelos de ciclo de vida tradicionais, pois não se trata efetivamente de um desenvolvimento interno de sistemas proprietários, mas sim da aquisição e adaptação de um sistema comercial desenvolvido externamente com objetivo de atender diversas empresas.

Figura 4 - Ciclo de vida de sistemas ERP



Fonte: Souza e Zwicker (2000, p. 50).

Levando em consideração dos conceitos de Sommerville (2011), Souza e Zwicker (2000) e Pfleeger (2014), podemos entender que o ciclo de vida do software comercial (ERP) e o ciclo de vida do software tradicional há algumas diferenças, abaixo o Quadro 1 mostra como o comparativo de processos de cada ciclo.

Quadro 1 - Ciclo de vida do software ERP x ciclo de vida do software tradicional

Processos	Ciclo de vida do Software ERP	Ciclo de vida do Software Tradicional
Definição de requisitos		x
Projeto de sistema de software		x
Decisão e seleção	x	
Implementação e teste	x	x
Entrega e integração	x	x
Operação e manutenção,	x	x

Fonte: elaborado pelo autor (2019).

Conforme Sommerville (2011), a implantação de sistemas ERP envolve diversos pontos, entre eles, a transferências de dados existentes no sistema utilizado pelo cliente, esses dados podem estar armazenados em diversos formatos, mas o presente estudo se limitará a falar do armazenamento em bancos de dados relacionais conforme será visto na seção a seguir.

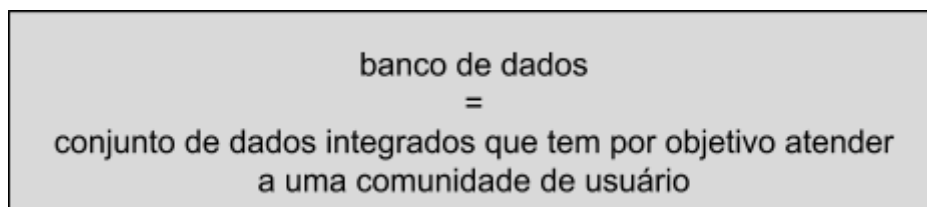
2.2 Banco de dados

Para MACHADO (2014), um banco de dados é um conjunto de dados devidamente relacionados. De acordo com DATE (2003, p. 6) “Banco de dados é basicamente um sistema computadorizado de manutenção de registro; em outras palavras, é um sistema computadorizado cuja finalidade geral é armazenar informações. Ainda para DATE (2004, p. 10) “um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa”.

Segundo VICCI (2014, p. 7) “Os bancos de dados desempenham um papel crítico em quase todas as áreas em que os computadores são usados. Os sistemas de banco de dados surgiram em resposta aos métodos iniciais de gestão informatizados de dados comerciais” SILBERSCHATZ (2012, p. 2). Um banco de dados é projetado, construído e populado com dados para uma finalidade específica. Ele possui um grupo definido de usuários e algumas aplicações previamente concebidas nas quais esses usuários estão interessados. Diante dos conceitos citados, podemos definir que o banco de dados serve para armazenar dados, que

organizados conseguimos extrair as informações das organizações. Esta definição pode ser exemplificada na Figura 5.

Figura 5 - Conceitos de Banco de Dados.

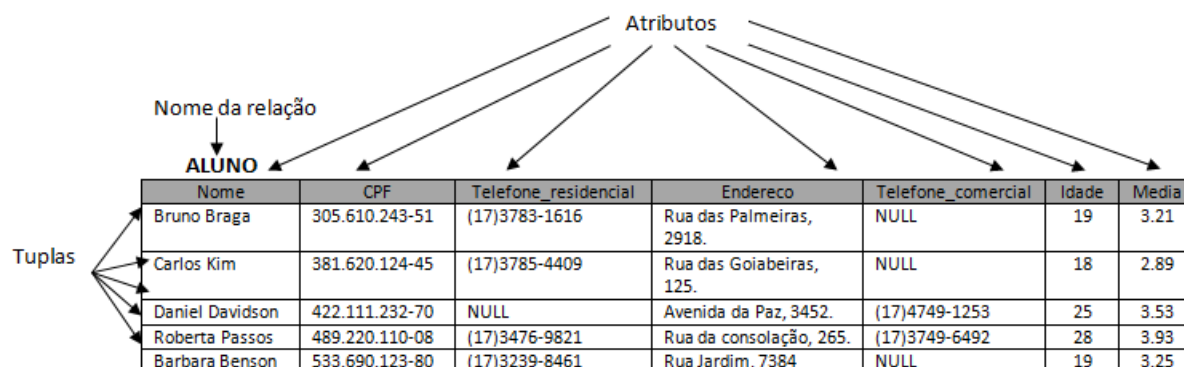


Fonte: HEUSER (2009, p. 22).

2.2.1 Estrutura do banco de dados

Um banco de dados é estruturado em tabelas (relação), linhas (tuplas) e colunas (atributos), conforme pode ser visto pela Figura 6.

Figura 6 - Estrutura do Banco de Dados.



Fonte: Elaborado pelo autor, adaptado de Elmasri (2011, p. 40).

2.2.1.1 Tabela / Relação

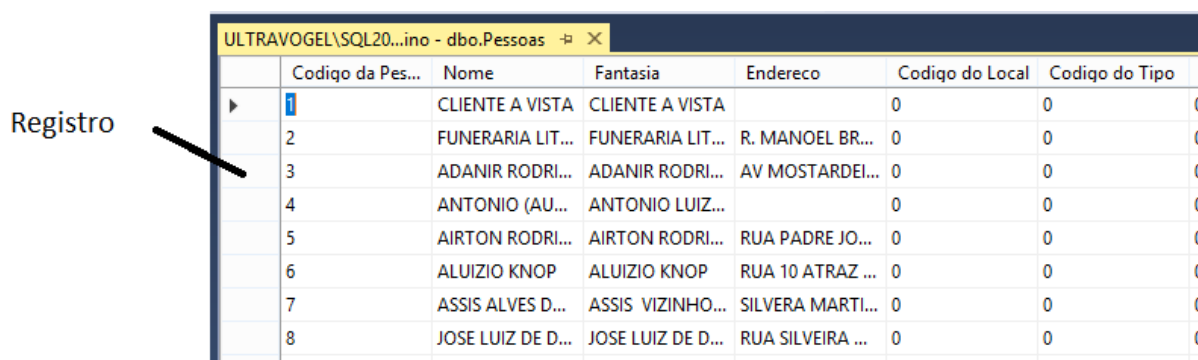
Segundo Heuser (2009, p. 120), “uma tabela é um conjunto não ordenado de linhas (tuplas, na terminologia acadêmica)”. Segundo Medeiros (2013), tabela é basicamente uma lista e possui um nome identificador próprio. Um banco de dados pode ser constituído de várias tabelas e estas possuem uma estrutura padronizada

compostas por atributos ou campos. Já Elmasri (2005) resume que relação é um conjunto de tuplas.

2.2.1.2 Linhas / Tuplas

Para ELMASRI (2005, p. 92) “Uma tupla pode ser considerada conjunto de pares (<atributos>, <valor>)”. Segundo Alves (2014), um registro é o conjunto de campos com valores de uma tabela. É a unidade básica para o armazenamento e recuperação de dados e que identifica a entrada de um único item de informação em particular numa tabela do banco de dados. São também chamados de tuplas ou n-uplas. Numa tabela cujos registros são formados por cinco campos, cada registro é denominado de 5-upla (ou quintupla). Esta definição pode ser exemplificada na Figura 7.

Figura 7 - Representação de registro.



The image shows a screenshot of a SQL Server Enterprise Manager window displaying a table named 'Pessoas' from the 'dbo' database. The table has 8 columns: 'Codigo da Pes...', 'Nome', 'Fantasia', 'Endereco', 'Codigo do Local', 'Codigo do Tipo', and 'C'. The first row is highlighted, and an arrow points to it with the label 'Registro'.

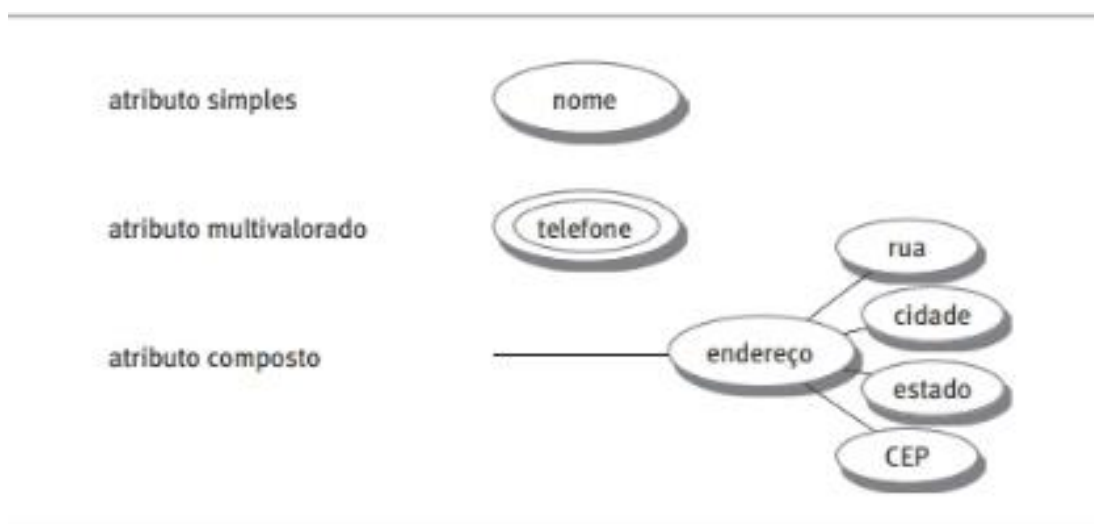
	Codigo da Pes...	Nome	Fantasia	Endereco	Codigo do Local	Codigo do Tipo	C
1		CLIENTE A VISTA	CLIENTE A VISTA		0	0	0
2		FUNERARIA LIT...	FUNERARIA LIT...	R. MANOEL BR...	0	0	0
3		ADANIR RODRI...	ADANIR RODRI...	AV MOSTARDEI...	0	0	0
4		ANTONIO (AU...	ANTONIO LUIZ...		0	0	0
5		AIRTON RODRI...	AIRTON RODRI...	RUA PADRE JO...	0	0	0
6		ALUIZIO KNOP	ALUIZIO KNOP	RUA 10 ATRAZ ...	0	0	0
7		ASSIS ALVES D...	ASSIS VIZINHO...	SILVERA MARTI...	0	0	0
8		JOSE LUIZ DE D...	JOSE LUIZ DE D...	RUA SILVEIRA ...	0	0	0

Fonte: Elaborado pelo autor, adaptado de ALVES (2014, p. 35).

2.2.1.3 Coluna / Atributo

Os atributos são específicos de cada entidade. Eles podem ser de diversos tipos, como exemplos podemos citar o simples, o composto e o multivalorado. O atributo simples contém um único valor para cada elemento da entidade. O atributo multivalorado permite conter informações com diversos valores. Outro tipo de atributo é o composto, que nos permite indicar um atributo que pode ser dividido em outros CARDOSO (2012), como pode ser visto na Figura 8.

Figura 8 - Tipos e atributos.



Fonte: Cardoso (2012, p. 30).

2.2.2 Dados

Para Santos (2009, p. 92), “dados são a matéria-prima para elaboração da informação. É representado por um conjunto de caracteres, dígitos ou símbolos que, tomados isoladamente, não transmite nenhum conhecimento, não contém um significado intrínseco.”

De acordo com Laudon e Laudon apud Rosini e Palmisano (2012, p. 4), dados são um “elemento que representa eventos ocorridos na empresa ou circunstâncias físicas, antes que tenham sido organizados ou arranjados de maneira que as pessoas possam entendê-los e usá-los”. Ou seja, os dados se referem a uma definição inicial de objetos, fatos, atividades e transações que são classificados e memorizados, porém não chegam a ser estruturados de forma a transmitir algum significado específico.

Levando em conta que cada sistema ERP possui uma estrutura de banco de dados diferente, quando realizamos a troca do sistema precisamos efetuar a migração dos dados entre os bancos de origem para o destino e para isso existem estratégias e metodologias que serão apresentadas na próxima seção.

2.3 Migração de dados

Segundo Mendonça (2009), o processo de migração de dados é basicamente o planejamento do projeto, a definição do escopo, a extração, a transformação (para o formato adequado), a transferência e a verificação dos dados. Ainda segundo o autor, esse processo de migração de enormes volumes de dados, não ocorre rotineiramente, por isso o setor de TIC (Tecnologia da informação e comunicação) das empresas não detêm experiência neste tipo de operação. De acordo com Mendonça (2009 p. 23), “Para realizar um projeto de migração de dados, é necessário que uma estratégia de migração seja adotada no início do projeto, já que seu planejamento e as ações necessárias dependem dessa estratégia”.

2.3.1 Estratégias

A migração de dados possibilita que as informações sejam transformadas e se tornem compatíveis para serem inseridas em um novo sistema. Entretanto esse processo deve ser realizado com eficiência, prudência e inteligência, para assegurar que nenhuma informação seja perdida no processo. Existem duas estratégias conhecidas de migração a *Big Bang* e *Trickle* (ROMAN e NOTARI, 2018).

Mendonça (2019) explica e exemplifica no Quadro 2 as estratégias de migração de dados *Big Bang* e *Trickle*.

- *Big Bang* - Este tipo de migração é caracterizada por ser realizada em uma vez só. Neste caso, o sistema original deve ficar fora do ar enquanto os dados são extraídos, transformados e carregados na aplicação destino. A grande vantagem desta estratégia é ser finalizada no menor tempo possível. Entretanto, ela apresenta alguns pontos negativos, essa estratégia não se aplica a organizações que não podem ter o seu sistema fora do ar por muito tempo, há uma grande pressão sobre a execução da migração e a realização de seus testes. Normalmente, quando esta estratégia é adotada, o processo de execução da migração ocorre durante um fim de semana ou feriado;
- *Trickle* - As migrações que optam pela estratégia *Trickle* são executadas de forma incremental. Os dois sistemas executam em paralelo e os dados são migrados em partes. Isto pode ser desenvolvido utilizando processos em tempo

real para transferir os dados da base de origem para a base destino e para manter os dados da base destino atualizados.

Quadro 2 - Estratégias de migração de dados

Características	Big Bang	Trickle
Necessidade de sincronismo	Não	Sim
Complexidade	Média	Alta
Sistema legado fora do ar	Sim	Não
Dados migram primeiro	Sim	Não
Porte da migração	Pequeno	Médio / Grande
Interoperabilidade entre as aplicações	Não	Sim
Compatibilidade com as metodologias de migração de sistemas legados	Butterfly	Chicken Little

Fonte: Mendonça (2009, p. 9).

2.3.2 Metodologias

Para que a migração dos dados ocorra de uma forma que mantenha a integridade e funcionalidade do sistema ERP, serão apresentadas três metodologias de migração de dados: *Chicken Little*, *Butterfley* e ETL (*Extract, Transform, Load*).

2.3.2.1 Chicken Little

É uma metodologia de migração que se caracteriza por ser incremental, usada muitas vezes para migração de sistemas legados onde os dados são migrados à medida que o novo sistema avança, até que ele contemple todas as funcionalidades do sistema legado (OLIVEIRA e MARCELINO, 2012).

A estratégia de migração *Chicken Little*, consiste em onze pequenos passos, que são realizados de forma iterativa, em que a migração se torna realizável, uma vez que a migração real é dividida em partes menores (princípio *Divide and Conquer*).

Chicken Little ainda significa uma reformulação completa do sistema. BROY 2005 (apud OLIVEIRA e MARCELINO, 2012) definem os onze pequenos passos de estratégia de migração:

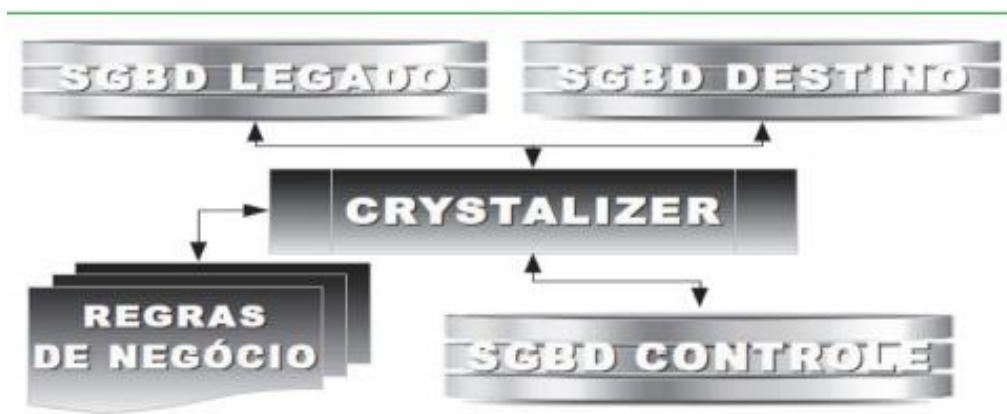
1. Analisar o sistema legado;
2. Decompor a estrutura do sistema legado;
3. Projetar a interface destino;
4. Projetar a aplicação destino;
5. Projetar a base de dados destino;
6. Instalar o ambiente destino;
7. Criar e instalar os gateways necessários;
8. Migrar as bases legadas;
9. Migrar as aplicações legadas;
10. Migrar as interfaces legadas;
11. Mudar para o sistema destino.

2.3.2.2 Butterfley

A metodologia de migração de banco de dados denominada *Butterfley*, exemplificada na Figura 9, é caracterizada por não ter a interoperabilidade entre as aplicações, isto é, o sistema legado permanece em operação, enquanto o novo sistema pode ser desenvolvido e testado previamente antes da migração, sem causar transtornos e sem afetar o funcionamento do software (OLIVEIRA e MARCELINO, 2012).

Sua abordagem é muito semelhante à ETL, com a diferença que esta abordagem propõe o desenvolvimento de um sistema intermediário de migração de dados chamado *Christalizer*, que tem como função converter o modelo do banco de dados, transpondo a informação e realizando os ajustes necessários, de um SGBD para o SGBD destino, por meio de regras de negócio campos que não podem ser nulos ou que serão completados por uma consulta no destino incluídas manualmente pelo administrador do banco de dados Andrade apud OLIVEIRA E MARCELINO, (2012, p 27) .

Figura 9 - Estratégia de migração Butterfley



Fonte: Oliveira e Marcelino (2012, p. 190).

2.3.2.3 ETL

O processo de ETL destina a extração, transformação e carga de um ou mais bancos de dados de um ou mais bancos de dados de origem para um ou mais bancos de dados de destino, sendo que a extração e carga são obrigatórios e a transformação é opcional (ABREU, 2010).

Ribeiro (2010, p. 2) afirma que “ETL é aplicado em migração de dados para sistemas de informação, *business intelligence* e aplicações de *data warehouse*, *data marts*”.

O ETL é composto por três etapas: extração (*extract*), transformação (*transform*) e carga (*load*).

- Extração: Haberkorn (2009) diz que está é a etapa de extrair os dados da base ERP ou de outras bases existentes na empresa, já Ribeiro (2010) complementa que é nessa fase que identificamos o tipo de armazenagem, estrutura e modelagem de dados a serem extraídos, como também a necessidade de utilizar uma ferramenta para extrair os dados de origem para o destino.
- Transformação: é nesta etapa que efetuamos a transformação dos dados em um mesmo padrão, pois na base de dados de origem eles podem estar em unidades de medidas, padrões e regras de negócios diferentes que a base de dados de destino, conforme Haberkorn (2009) e Abreu (2010);
- Carga: esta é a última etapa do ETL, que é a gravação dos dados extraídos, tratados e transformados da sua origem para o banco de dados de destino (HABERKORN, 2009) e (ABREU, 2010).

3 METODOLOGIA

Conforme Vergara (2006, p. 12), “método é um caminho, uma forma, uma lógica de pensamento”. Assim, este capítulo tem por objetivo detalhar os procedimentos que norteiam o desenvolvimento deste estudo, bem como apresentar conceitos e técnicas utilizadas no seu desenvolvimento.

3.1 Caracterização da pesquisa

3.1.1 Pesquisa bibliográfica

Para Marconi e Lakatos (2010) é o levantamento da bibliografia já publicada que tenha relação com o tema. Tem como objetivo colocar o pesquisador em contato direto com tudo o que foi escrito sobre determinado assunto, como livros, revistas, artigos, jornais, dentre outros. Para os autores Cervo; Bervian e Silva (2007), a pesquisa bibliográfica pretende explicar uma situação através de contribuições culturais ou científicas já divulgadas sobre o tema ou assunto e com isso contribuir para o desenvolvimento da ferramenta desenvolvida no estudo.

3.1.2 Pesquisa exploratória

O estudo é de caráter exploratório e tem como finalidade desenvolver uma ferramenta para auxiliar no processo de implantação de um novo sistema ERP. Para Gil (1996) uma pesquisa exploratória consiste em tornar um problema mais claro, possibilita que o pesquisador tenha mais familiaridade com o problema. Acrescenta que a caracterização da pesquisa exploratória ocorre quando envolve o levantamento

bibliográfico e entrevistas com os profissionais envolvidos no processo. Vergara (2006, p. 47), acredita que a pesquisa exploratória

[...] não deve ser confundida com leitura exploratória, é realizada em área no qual há pouco conhecimento acumulado e sistematizado. Por sua natureza de sondagem, não comporta hipóteses que, todavia, poderão surgir durante ou ao final da pesquisa.

3.1.3 Pesquisa quantitativa

Este tipo de pesquisa, de acordo com Roesch (2013), deve ser utilizada quando o principal enfoque do estudo é mensurar as relações entre as variáveis ou mesmo avaliar o resultado de um sistema ou projeto. O autor acrescenta que o método quantitativo utiliza principalmente dados padronizados que possibilitam ao pesquisador preparar sumários, fazer comparações e generalizações. Diante destes conceitos o presente estudo se caracteriza como pesquisa quantitativa por realizar comparações entre o processo atual de migração de dados e o processo a ferramenta desenvolvida.

3.1.4 Pesquisa experimental

O presente estudo caracteriza também como pesquisa experimental por buscar realizar testes em laboratório, para identificar quais pontos tiveram melhora em desempenho ao utilizar a ferramenta desenvolvida, segundo Gil (2009, p. 27).

A pesquisa experimental constitui o delineamento mais prestigiado nos meios científicos. Consiste essencialmente em determinar um objeto de estudo, selecionar as variáveis capazes de influenciá-lo e definir as formas de controle e de observação dos efeitos que a variável produz no objeto. Trata-se, portanto, de uma pesquisa em que o pesquisador é um agente ativo, e não um observador passivo.

A pesquisa experimental tem como objetivo proporcionar familiaridade com o problema, tornando mais explícito ou a construir hipóteses. Seu planejamento pode ser bastante flexível, pois considera variados aspectos relativos ao fato ou fenômeno estudado (GIL, 2009). Ainda para o autor, no modelo clássico de pesquisa experimental, é necessário que o pesquisador manipule pelo menos um dos fatores que se acredita ser responsável pela ocorrência do fenômeno que está sendo pesquisado.

3.2 Ferramentas

Nesta seção serão abordadas as principais ferramentas utilizadas na construção da solução proposta.

3.2.1 Linguagem de programação Java

O presente estudo utilizará a Linguagem de Programação Java para o desenvolvimento da ferramenta. Jandl (2013, p. 17) afirma que:

[...] é um ambiente completo de desenvolvimento e execução de programas que reúne um conjunto ímpar de facilidades: uma linguagem completamente orientada a objetos, robusta, muito portátil, que permite operação em rede (com destaque à internet), a distribuição de aplicações e que incorpora diversas características voltadas à segurança.

A seguir, algumas características da Linguagem de Programação Java:

- Orientada a objetos: O Java é uma linguagem de programação totalmente orientada a objeto, oferecendo mecanismos de abstração, encapsulamento e hereditariedade (JANDL, 2013);
- Independência de plataforma: O Java não é compilado para uma plataforma específica, mas sim são gerados *bytecodes*, que é uma forma intermediária para ser interpretado pela JVM (Java Virtual Machine) que é o interpretador de *bytecodes*, implementado para qualquer plataforma (JANDL, 2013);
- Performance: com os atuais compiladores JIT (Just In Time) que transformam os códigos da JVM em códigos nativos, a performance do Java é comparada a programas nativos HORSTMANN (2010) e JANDL (2013);
- Segurança: como é possível obter aplicações por meio de uma rede o Java tem mecanismos de segurança JANDL (2013);
- Multithreaded: O Java tem suporte a *multithreaded*, e segundo HORSTMANN (2010) o Java simplesmente transfere a implementação para o sistema operacional.

3.2.2 Netbeans 8.2

O NetBeans foi o IDE (*Interface Development Environment*) de desenvolvimento escolhida para o desenvolvimento da ferramenta, segundo

Gonçalves (2006, p. 23) "o NetBeans é um a um ambiente de desenvolvimento, *open-source* escrito totalmente em Java."

3.2.3 Banco de dados PostgreSQL

O PostgreSQL foi banco de dados escolhido para ser responsável por armazenar informações pertinentes à ferramenta por ser um banco de dados compatível com os principais sistemas operacionais disponíveis atualmente como: Windows, Linux, MacOS, Unix (MILANI, 2008).

Conforme Gonzaga (2007, p. 4) "O PostgreSQL é robusto, confiável, rico em recursos, implementa diversas características de orientação a objetivos e suporte grande parte do SQL:2003.", além disso possui bibliotecas e *drivers* de conexão a diversas linguagens de programação como: C, Java, PHP, Python Javascript, C#, VB.NET, entre outras (MILANI, 2018).

4 TRABALHOS RELACIONADOS

Antes de iniciar o estudo deste o trabalho de conclusão, foram pesquisados trabalhos já realizados sobre o tema, alguns estudos foram encontrados e serviram de base para formar a ideia da criação da ferramenta desenvolvida. Neste capítulo serão apresentados os trabalhos relacionados e o que cada um contribuiu para o desenvolvimento deste projeto.

4.1 Análise comparativa das ferramentas de ETL – Kettle e Talend

O trabalho de análise comparativa das ferramentas de ETL – Kettle e Talend (LIRA FILHO, 2013) foi elaborado para obtenção do título de Bacharel no Curso de Sistemas de Informação da Universidade Federal da Paraíba, com o propósito de avaliar e comparar as ferramentas de ETL Kettle e Talend no cenário de migração de dados.

Nesta análise comparativa entre as duas ferramentas, o autor se baseou em critérios pré-definidos, subdividido em três categorias: a forma que realizam as transformações, as funcionalidades existentes e o desempenho das ferramentas. Por fim o autor concluiu que as ferramentas têm características distintas, sendo que a ferramenta Kettle tem uma maior flexibilidade na parte de transformações e funcionalidades, já a Talend teve uma performance e uso de processamento melhor.

Para o estudo apresentado nesse trabalho, constatou-se que as ferramentas existentes são complexas e geralmente são utilizadas para migrações de maior porte, em cenários de *data warehouse* e *business intelligence*.

4.2 Processos de implantação e migração de dados com utilização de ETL para um ERP comercial

O artigo de processos de implantação e migração de dados com utilização de ETL para um ERP comercial (RIBEIRO, 2010) foi elaborado por um estudante do curso de Sistemas de Informação - Campus Canoas da Universidade Luterana do Brasil (Ulbra), com o objetivo criar um processo automatizado para implantação de um sistema ERP comercial com foco na migração de dados utilizando a metodologia ETL.

Para o presente estudo este trabalho serviu para reforçar a interpretação obtida no capítulo anterior referente às ferramentas disponíveis. Além disso, o autor desenvolveu uma ferramenta que faz a manipulação das informações da origem sobre fontes simples de dados, que são arquivos textos e planilhas eletrônicas no padrão Microsoft Office Excel.

O autor relatou como melhoria e trabalho futuro, a possibilidade de a ferramenta acessar outras fontes de dados, com conexão direta aos próprios bancos de dados de origem, facilitando a extração dos dados. A partir disso surgiu a ideia para o desenvolvimento deste trabalho de conclusão.

4.3 Requisitos para ferramentas de migração de dados

O artigo requisitos para ferramentas de migração de dados (SANTOS e RODRIGUES, 2013) foi elaborado por dois estudantes da Universidade Federal de Piauí, com o objetivo de definir requisitos que devem ser levados em consideração, com base no estudo dos autores, para ferramentas de migração de dados, como, por exemplo, os requisitos de possibilitar migração de estruturas diferentes, migração incremental, bases com modelagens diferentes, testabilidade e usabilidade. Com isso os autores escolheram algumas ferramentas para servir como base experimental para avaliação dos requisitos do trabalho.

Os autores concluíram que há várias ferramentas não atendem os requisitos definidos no trabalho e afirma que o estudo apresentado facilitará o desenvolvimento e criação de novas ferramentas de migração de dados.

Com esta afirmação, este trabalho relacionado serviu para nortear o desenvolvimento da ferramenta deste estudo, ajudando a definir alguns requisitos e

eliminar outros, para que o trabalho não se estendesse mais que o tempo disposto para o estudo.

4.4 Migração e testes de um ERP do SGDB SQL Server para o SGDB PostgreSQL

O trabalho migração e testes de um ERP do SGDB SQL Server para o SGDB PostgreSQL (SELZLEIN, 2009) foi elaborado por um estudante da Universidade de Caxias do Sul para obtenção do título de Bacharel no Curso de Sistemas de Informação, com o propósito de estudar a migração do SGDB (Sistema de Gerenciamento de Banco de Dados) de uma empresa de software e avaliar qual dos dois atende mais as necessidades dos seus clientes.

Neste trabalho exigiu com que o autor migrasse os dados do SQL Server para o PostgreSQL e com isso o autor se utilizou de uma ferramenta existente para a migração dos dados onde percebeu a necessidade de desenvolvimento e adaptações a parte da ferramenta. A ferramenta chamada de *Migrate Database* que tem o objetivo de ser uma ferramenta genérica para migração de bancos relacionais, mas que acabou não contemplando algumas características importantes como, por exemplo, chaves estrangeiras.

Por mais que o objetivo fim do estudo deste autor não teve relação direta com este trabalho de conclusão, o mesmo serviu para avaliar os problemas reportados pelo autor no decorrer da troca de informações entre os bancos de dados e conectividade com os bancos de dados, como, por exemplo, o volume excessivo de drivers a serem baixos pelo autor.

5 IMPLEMENTAÇÃO DA FERRAMENTA

O presente estudo propôs o desenvolvimento de uma ferramenta para auxiliar na migração de bancos de dados entre sistemas ERP, tendo seu público alvo pequenas e médias empresas de Tecnologia da Informação que efetuam migrações de sistemas ERP. Neste cenário, as validações foram realizadas partindo do pressuposto que as bases de dados de origem são diferentes, mas a base de destino é conhecida, ou seja, sempre a mesma. Portanto a utilização da ferramenta agilizar as conexões entre os bancos, avaliação de tabelas e campos do banco de dados de origem, visualização dos dados e a realização de vínculos e criação de *scripts* para efetuar a migração dos dados.

São vários os aspectos que fazem com que o tema de migração de banco de dados seja amplamente discutido e estudado, um deles é a segurança das informações do cliente que precisa ser levada muito em consideração, como também a diversidade de banco de dados diferentes disponíveis. Conforme visto no capítulo anterior, existem estudos sobre ferramentas prontas disponíveis no mercado como também propostas de novos desenvolvimentos para melhoria deste processo, com isso, este estudo propôs complementar os estudos anteriores e desenvolver a ferramenta utilizando a estratégia de migração *Big Bang* e a metodologia de migração ETL (*Extract, Transform, Load*).

Portanto na experimentação de validação desse trabalho, realizou-se um comparativo de três migrações de dados realizadas anteriormente pelo autor, que foram previamente documentadas, sem a utilização de nenhuma ferramenta específica para isso e posteriormente realizadas as mesmas migrações utilizando a ferramenta desenvolvida, por fim foram avaliados os resultados. Para um melhor

entendimento destes cenários na seção a seguir será explicada como foram realizadas as três migrações iniciais sem a ferramenta e seus respectivos bancos de dados de origem.

5.1 Cenário avaliados

Cada cenário avaliado teve seu tempo previamente documentado pelo autor e serviu como base para realizar a análise comparativa entre o cenário anterior com o cenário após a utilização da ferramenta.

Em todos os cenários, a avaliação de dados foi manual, não utilizando nenhuma ferramenta específica para o apoio das exportações, transformações e carga de dados.

- Cenário A: neste cenário o banco de dados utilizado pelo ERP anterior do cliente era o Postgres. O volume de dados desta base era significativo, mas possuía uma estrutura de dados de fácil entendimento, além do cliente conhecer a estrutura de tabelas;
- Cenário B: neste cenário o banco de dados utilizado pelo ERP anterior do cliente era o Firebird. A base não possuía um volume significativo de informações, mas possuía vários erros de estrutura que dificultou a migração;
- Cenário C: neste cenário o banco de dados utilizado pelo ERP anterior do cliente era o SQL Server. Neste caso teve um volume significativo de informações a serem migradas, mas pelo banco de dados de origem e destino serem iguais facilitou a criação de *scripts*.

5.2 Modelagem da ferramenta

Com base nos três cenários descritos foram levantadas as demandas que a ferramenta desenvolvida precisaria realizar para que atingisse o objetivo principal do estudo - tornar o processo de migração de dados mais ágil, eficiente e eficaz.

5.2.1 Levantamento de requisitos

Nesta seção foram descritos os requisitos da ferramenta. Estes requisitos são divididos em funcionais e não funcionais e são apresentados nas próximas seções. Os requisitos foram classificados com base em prioridades, conforme pode ser observado no Quadro 3.

Quadro 3 - Prioridades dos requisitos

Prioridade	Descrição	Abreviaturas
Obrigatório/Essencial	Requisitos que o sistema deve atender	O
Importante	Requisitos que o sistema deveria atender	I
Desejável	Requisitos que se for possível, o sistema poderia atender	D

Fonte: elaborado pelo autor (2019).

5.2.1.1 Requisitos funcionais

Conforme Sommerville (2011), os requisitos funcionais são funcionalidades que o sistema deve obrigatoriamente possuir e como o sistema deve agir ao receber um conjunto de informações de entrada de dados. A seguir, o Quadro 4 demonstra os requisitos funcionais da ferramenta desenvolvida, sendo que todos os requisitos foram atendidos.

Quadro 4 - Requisitos funcionais

Nº	Descrição	Prioridade
001	Possibilitar visualizar o repositório de metadados da origem e destino em paralelo. A proposta é que o sistema exiba lado a lado cada estrutura de banco de dados com o objetivo de poder comparar informações da origem com o destino e fazer suas devidas ligações.	O
002	Possibilitar mapeamento de campos com tipos de dados diferentes. O sistema deve ser capaz de fazer ligações entre campos de tipos diferentes de dados, sempre respeitando o formato do campo destino e tratando as informações na migração.	I
003	Validar tamanho dos campos comparando origem com o destino. Ao mapear dois campos o sistema precisa ser capaz de fazer comparação se o dado que está na origem poderá ser migrado para o campo do destino.	D

004	Possuir um comparador entre conteúdo da origem com o destino. O objetivo é agilizar a comparação entre os dados da origem com o destino de uma maneira visualmente fácil.	D
005	Possibilitar armazenar a solução criada para utilizações e execuções futuras. O projeto é desenvolvido antes e executado somente no dia da migração e com isso a ferramenta precisa armazenar a solução montada pelo usuário e carregá-la em um segundo momento.	O
006	Possuir rotina de validação da solução, sendo capaz de validar a solução criada antes mesmo de executar ela no banco de dados de destino.	O
007	Possibilitar armazenar as amarrações (DE-PARA) feitas pelo usuário para serem executadas no dia da migração.	O
008	Manter cadastro de destino. Objetivo é manter o cadastro do banco de dados de destino.	O
009	Manter cadastro de origens. O objetivo é manter o cadastro dos bancos de dados de origem.	O
010	Possibilitar fazer DE-PARA entre tabelas e campos da origem para tabelas e campos de destino.	O
011	Efetuar migração dos dados da origem para o destino. O objetivo é rodar uma rotina que via <i>scripts</i> efetue a migração dos dados da origem para o seu destino.	O

Fonte: elaborado pelo autor (2019).

5.2.1.2 Requisitos não funcionais

Conforme Sommerville (2011), os requisitos não funcionais estão relacionados a características específicas do sistema como, por exemplo, confiabilidade, segurança e desempenho. A seguir o Quadro 5 demonstra os requisitos não funcionais da ferramenta desenvolvida.

Quadro 5 - Requisitos não funcionais

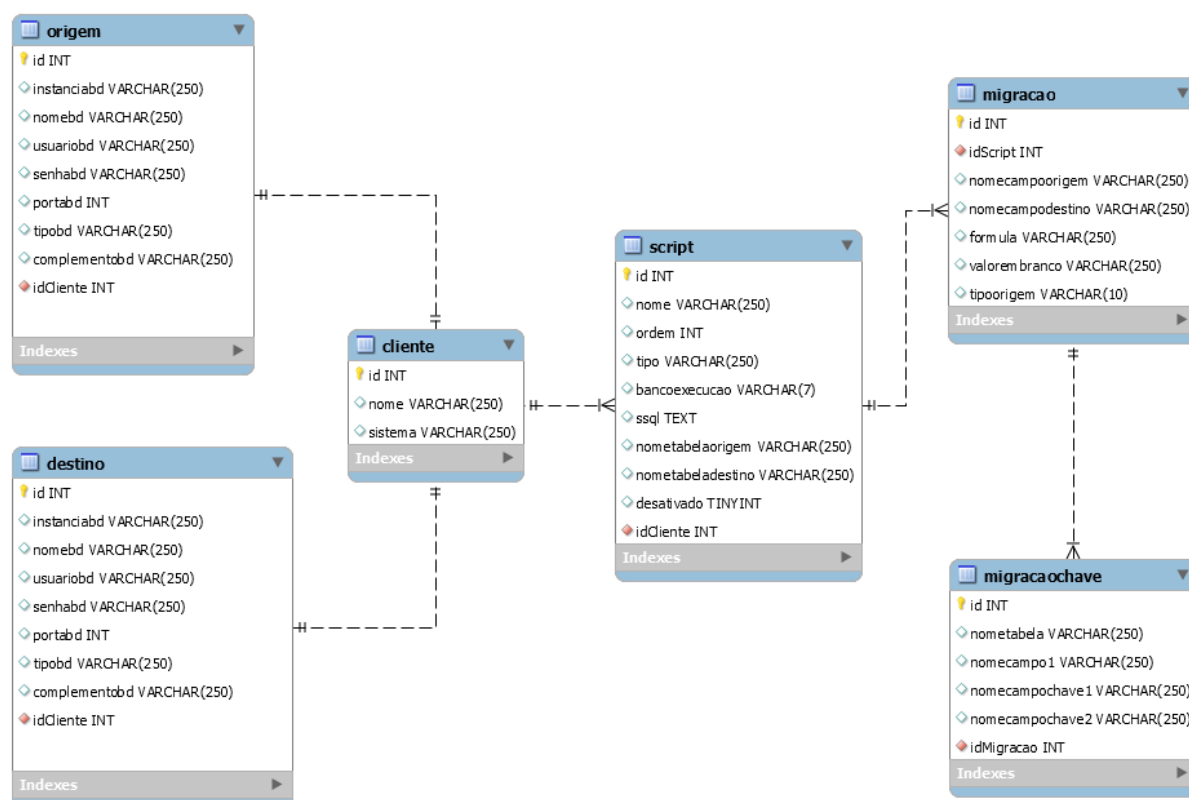
Nº	Descrição	Prioridade
001	Utilizar linguagem de Programação Java	O
002	Utilizar banco de dados PostgreSQL para armazenar a solução da aplicação	O
003	Possibilitar extrair (origem) e inserir (destino) dados de três gerenciadores de banco de dados diferentes: PostgreSQL, Firebird, SQL Server	O

Fonte: elaborado pelo autor (2019).

5.2.2 Modelo entidade-relacionamento

A ferramenta desenvolvida faz conexão e utilização dos bancos de dados de origem e destino, sendo que cada banco de dados terá seu modelo ER. Informações pertinentes à ferramenta são armazenados em um banco de dados conforme modelo ER apresentado na Figura 10.

Figura 10 - Modelo entidade-relacionamento



Fonte: elaborado pelo autor (2019).

Com o objetivo de complementar o modelo entidade-relacionamento da Figura 10, o Quadro 6 apresenta a descrição de cada uma das tabelas que integram este modelo.

Quadro 6 - Detalhamento das tabelas do banco de dados

Nome da Tabela	Descrição da Tabela
origem	Tabela com os dados de conexão do banco de dados de origem
destino	Tabela com os dados de conexão do banco de dados de destino

cliente	Tabela com dados do cliente / projeto
script	Tabela central que será usada para armazenar vínculos da tabela ou visão de origem com a tabela de destino, como também <i>scripts</i> de <i>update</i> e <i>delete</i>
migracao	Tabela central que será usada para armazenar vínculos do campo da origem para o campo de destino
migracaochave	Tabela central que será usada para armazenar vínculos o registro de migração for 1=N ou N=1, armazenando dados da tabela vinculada

Fonte: elaborado pelo autor (2019).

5.3 Desenvolvimento da ferramenta

Esta seção tem como objetivo demonstrar a ferramenta final desenvolvida, apresentando características de projeto, como também telas e funcionalidades que o usuário deve utilizar para criação dos projetos de migração de dados.

5.3.1 Classes e estrutura de dados

O estudo limitou-se a testar um conjunto de três tipos de bancos de dados diferentes para os dados de origem e um tipo de banco de dados para o recebimento das informações no destino. Mesmo com este escopo, foi investido um tempo para que a ferramenta fosse estruturada para agregar outros tipos de bancos de dados rapidamente ao projeto agilizando melhorias futuras.

As classes da ferramenta foram preparadas para receber novos tipos de bancos de dados tanto na origem quanto no destino possibilitado a evolução da ferramenta. Mesmo que seja necessário desenvolvimento para agregar outros *drivers* de conexão, padrões SQL entre outros, o código fonte estando estruturado de forma correta agiliza a evolução da ferramenta. A Figura 11 demonstra um trecho de código fonte que está preparado para recebimento de novos tipos de bancos de dados.

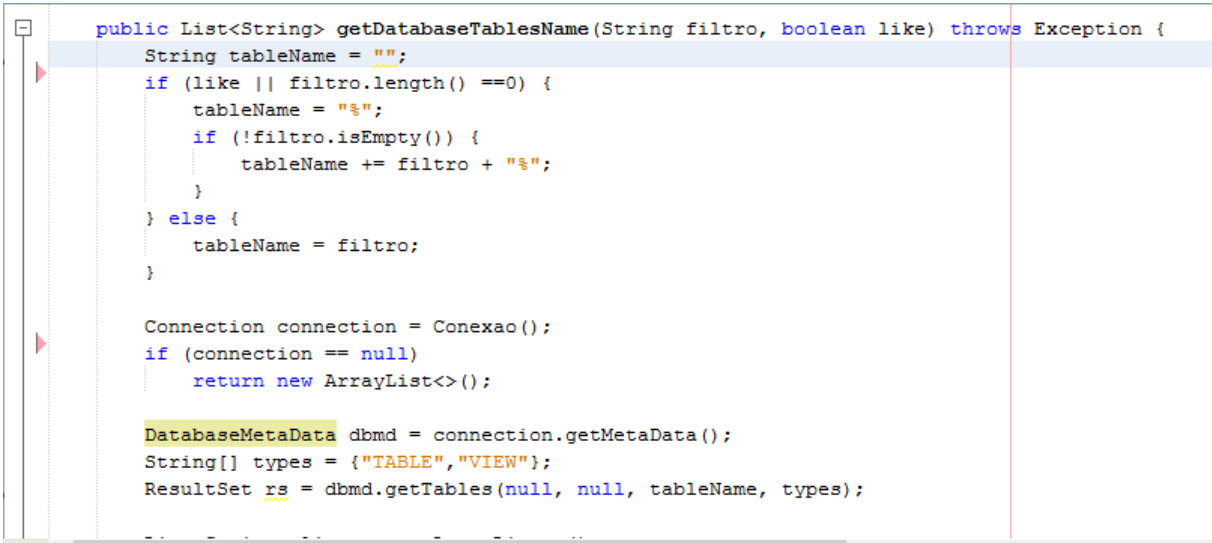
Figura 11 - Exemplo código fonte tipos de banco

```
2
3  import br.com.vogel.negocio.apoio.Strings;
4
5  public class DBUtil {
6      public static String obterNomeBancoPeloDriver(String driver) {
7          switch (driver) {
8              case DriverBD.FIREBIRD:
9                  return Strings.banco_firebird;
10             case DriverBD.POSTGRES:
11                 return Strings.banco_postgres;
12             default:
13                 return Strings.banco_sqlserver;
14             }
15         }
16     }
```

Fonte: elaborado pelo autor (2019).

A leitura e visualização da estrutura de tabelas e campos de cada banco de dados também teve um tratamento especial no desenvolvimento do projeto, para que não fosse necessário entender como cada gerenciador de banco de dados exibe a sua estrutura de dados. Foram utilizadas classes com padrões nativos da própria Linguagem de Programação Java, não necessitando desenvolver mais nada quando agregar outro tipo de banco de dados. Conforme exemplificado na Figura 12 a rotina em questão se utiliza do método `getMetaData()` do arquivo de conexão para fazer a leitura da estrutura do banco de dados.

Figura 12 - Exemplo código fonte leitura estrutura banco



```

public List<String> getDatabaseTablesName(String filtro, boolean like) throws Exception {
    String tableName = "";
    if (like || filtro.length() == 0) {
        tableName = "%";
        if (!filtro.isEmpty()) {
            tableName += filtro + "%";
        }
    } else {
        tableName = filtro;
    }

    Connection connection = Conexao();
    if (connection == null)
        return new ArrayList<>();

    DatabaseMetaData dbmd = connection.getMetaData();
    String[] types = {"TABLE", "VIEW"};
    ResultSet rs = dbmd.getTables(null, null, tableName, types);
}

```

Fonte: elaborado pelo autor (2019).

5.3.2 Telas e funcionalidades da ferramenta

Para que possamos entender e conhecer as funcionalidades da ferramenta desenvolvida esta seção demonstra cada tela juntamente com o detalhamento das suas funcionalidades, iniciando pela Figura 13 onde é demonstrada a tela inicial da ferramenta. Logo abaixo do rótulo [Meus Projetos] é possível visualizar uma listagem com os projetos já cadastrados, que é seguida de quatro botões, onde cada um deles executa as funcionalidades a seguir detalhadas. Os botões necessitam de um projeto selecionado na listagem, caso nenhum esteja selecionado a ferramenta gera um aviso e permanece na tela inicial.

- Selecionar: Ao clicar neste botão o sistema direciona o usuário para a tela de conexões passando o cliente para ela;
- Alterar: Ao clicar neste botão o sistema direciona o usuário para a tela de manter projetos, para que ele possa alterar informações referente ao projeto selecionado;
- Excluir: Ao clicar neste botão a ferramenta solicita confirmação se o usuário deseja realmente excluir o projeto selecionado, caso a resposta for positiva, é deletado o projeto no banco de dados a nível físico;

- Duplicar: Ao clicar neste botão a ferramenta solicita confirmação se o usuário deseja realmente duplicar o projeto selecionado, caso positivo duplica o mesmo com as todas as suas informações.

Figura 13 - Tela Inicial



Fonte: elaborado pelo autor (2019).

Ainda na Figura 13 também podemos visualizar na parte inferior da tela o menu principal contendo cinco botões de atalhos representados por ícones, que estão presentes em todas as telas da ferramenta. O nome da tela de cada atalho será apresentado a seguir seguindo a sequência da direita para a esquerda, já a funcionalidade de cada tela será demonstrada no decorrer do capítulo.

- Ícone Inicial;
- Ícone Manter Projetos;
- Ícone de Conexões;
- Ícone de *Scripts*;
- Ícone de Migração.

A tela de manter projetos que está sendo demonstrada na Figura 14 pode ser acessada pelo menu principal quando for necessário cadastrar um novo projeto ou pelo botão editar da tela inicial, representada na Figura 13, quando for preciso alterar informações de um projeto existente.

Figura 14 - Tela Manter Projetos



A interface da aplicação 'MigradorDB' apresenta uma tela para gerenciamento de projetos. No topo, há uma barra azul com o título 'MigradorDB' e ícones de minimizar, maximizar e fechar. O corpo da tela é branco e contém um formulário centralizado com os seguintes elementos:

- Um campo de texto rotulado 'Nome Projeto'.
- Um campo de texto rotulado 'Nome Sistema'.
- Um botão retangular rotulado 'Cadastrar'.

Na base da interface, há uma barra de navegação com cinco ícones: uma casa (representando o menu principal), um ícone de perfil de usuário com um sinal de mais (representando o perfil do usuário), um relâmpago (representando configurações ou ferramentas), um ícone de lista com uma caneta (representando a tela atual de 'Manter Projetos'), e um ícone de compartilhamento.

Fonte: elaborado pelo autor (2019).

Para o objetivo do estudo não se fez necessário criar um cadastro de projetos mais completo, mesmo assim, foi acrescentado como sugestão no capítulo de trabalhos futuros para que esse cadastro possa ser complementado nas próximas versões.

A ferramenta tem objetivo de seguir um fluxo de telas, portanto na tela inicial ao selecionar um projeto o usuário é direcionado automaticamente para a tela de conexões, representada na Figura 15, esta tela tem o objetivo de cadastrar as informações para conexão com o banco de origem e destino, além de ter as funcionalidades de testar as conexões e salvar as mesmas.

Figura 15 - Tela de Conexões

The screenshot shows a software window titled 'MigradorDB' with a subtitle 'TESTE SQL Server'. It features two identical connection configuration panels side-by-side. The left panel is labeled 'Conexão Banco Origem' and the right panel is labeled 'Conexão Banco Destino'. Each panel includes text boxes for 'Host' (filled with 'ultraoqefisq2017'), 'Nome' (filled with 'TCCOrigem' and 'TCCDestino' respectively), 'Porta' (filled with '1433'), 'Usuário' (filled with 'sa'), and 'Senha' (filled with 'om'). Below these is a dropdown menu for 'Tipo de banco' set to 'SQL Server'. A large empty text area labeled 'Saída' is at the bottom of each panel, with a 'Testar' button centered below it. At the very bottom of the window, a 'Salvar' button is centered, and a row of five icons (home, user, lightning bolt, document, and share) is displayed.

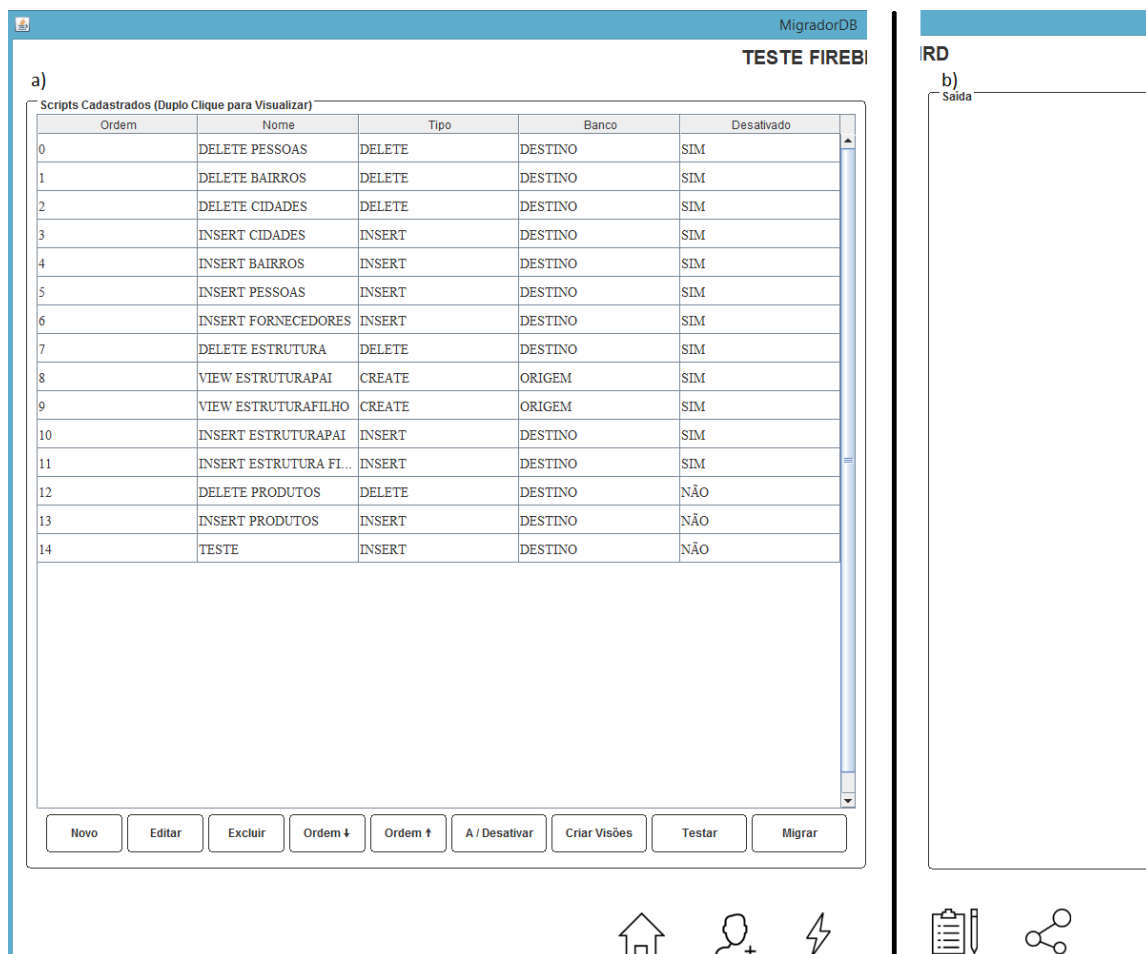
Fonte: elaborado pelo autor (2019).

Ao salvar as conexões do banco de dados, como visto na Figura 15, a ferramenta encaminhará o usuário para a tela de *scripts*. Esta tela foi dividida em duas partes, a primeira parte da tela, Figura 16(a), contém uma listagem com os *scripts* cadastrados e abaixo botões de atalhos onde cada botão contém uma funcionalidade, estas funcionalidades são:

- Novo: é a ação para cadastrar um novo *script*, nesta ação a ferramenta solicita o tipo de *script* a ser criado e abre a tela de migração;
- Editar: a ferramenta avalia se tem *script* selecionado na listagem, caso afirmativo, abre a tela de migração trazendo as informações já armazenadas do *script* selecionado;
- Excluir: a ferramenta avalia se tem *script* selecionado na listagem, caso afirmativo, pergunta ao usuário se ele deseja realmente excluir o *script*, se sim, exclui o *script* selecionado;
- Ordem ↑ e Ordem ↓: a ordem dos *scripts* para a migração dos dados é fundamental devido ao relacionamento entre as tabelas, estes dois botões reorganizam a ordem dos *scripts* selecionados para cima ou para baixo;
- A / Desativar: ativa ou desativa o *script* selecionado, fazendo a operação inversa do estado atual do *script*. Esta funcionalidade foi criada com o objetivo de executar parcialmente os *scripts* do projeto;

- Criar Visões: esta funcionalidade avalia todos os *scripts* que são criações de visões no banco de dados de origem e os executa para que o usuário tenha acesso as visões para criação de novos *scripts*;
- Testar: tem objetivo de testar todos os *scripts* ativos do projeto, a funcionalidade faz um backup do banco de dados de destino, restaura o mesmo, executa os *scripts* e depois deleta o banco de testes. Detalhes da rotina será demonstrada na seção 5.3.3;
- Migrar: tem o objetivo de realizar a migração final dos *scripts* ativos no banco de dados de destino. Detalhes da rotina será demonstrada na seção 5.3.3.

A segunda parte da tela de *scripts*, representada na Figura 16(b), é responsável por exibir todo e qualquer tipo de saída de informação que a ferramenta gere nesta tela, essa saída é apresentada em ordem cronológica contendo data e hora.

Figura 16 - Tela de *Scripts*

Fonte: elaborado pelo autor (2019).

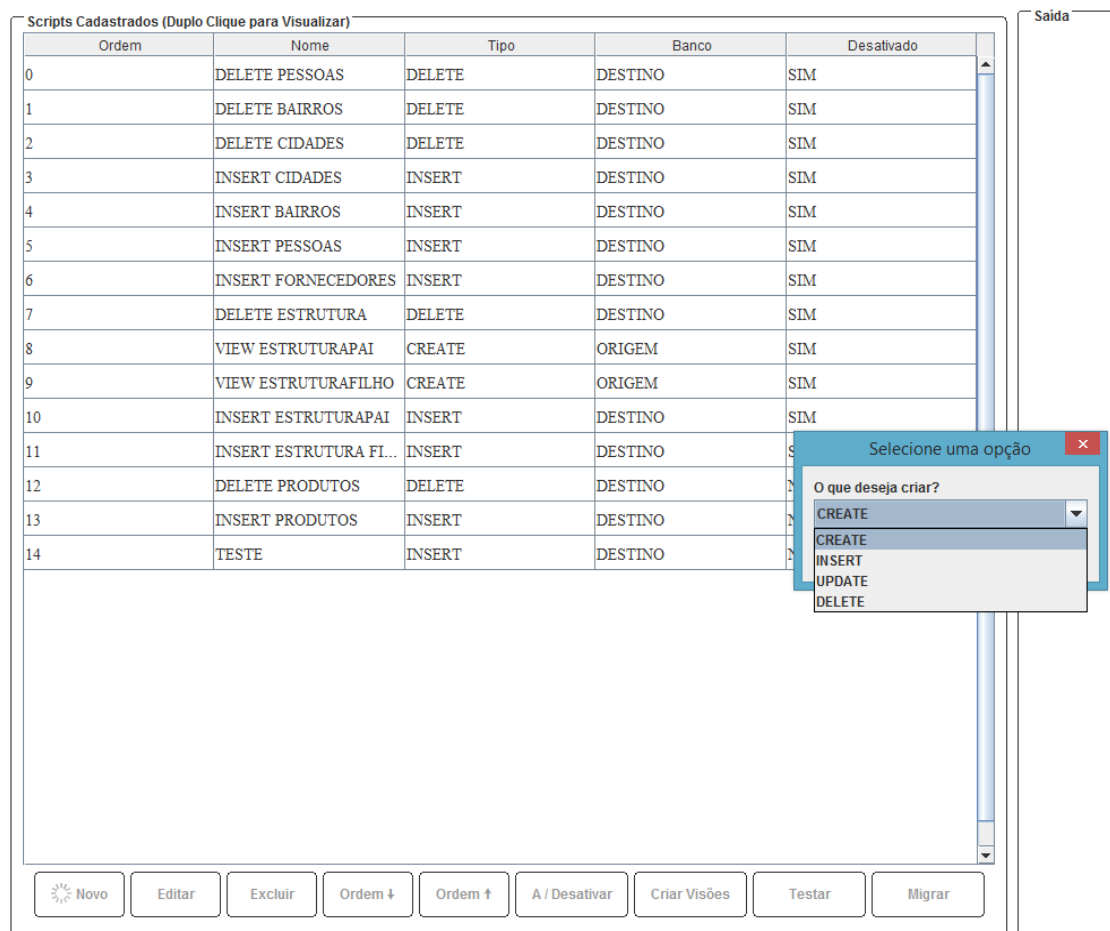
A partir da tela apresentada na Figura 16, alguns conceitos e regras de negócio importantes da ferramenta são abordados. O primeiro deles é a ordem de cada *script* que é preciso ser definida pelo usuário da ferramenta mediante o relacionamento e entendimento das tabelas em questão. Outro conceito são os tipos de *scripts* que foram divididos com o intuito de facilitar e agilizar a programação das rotinas de migração, os tipos de *script* são:

- *Create*: criação de consultas no banco de dados de origem, usado geralmente para montar algum tipo de consulta para facilitar os *scripts* de *insert*;
- *Delete*: criação de comandos de exclusão a serem executados no banco de dados de destino, geralmente usados para algum tipo de limpeza;

- *Update*: criação de comandos de atualização a serem executados no banco de dados de destino, geralmente usados para organizar e atualizar alguma informação após a migração;
- *Insert*: é a regra principal da ferramenta, nesta opção o usuário tem possibilidade de criar *scripts* para realizar a transferência dos dados do banco de dados de origem para o destino.

No cadastro *scripts* há a informação do banco de dados de destino onde o *script* será executado, essa informação é inserida automaticamente pela ferramenta com base em uma regra de negócio. Quando o tipo do *script* for *create* o *script* sempre será executado sobre o banco de dados de origem, criando as consultas e no final de cada processo sempre excluindo as mesmas do banco de dados de origem. Já os outros tipos serão executados sobre o destino, que seguirá os conceitos de cada funcionalidade (Testar e Migrar) já descrita acima. A seleção do tipo de *script* sempre é feita ao inserir um novo *script*, como pode ser verificado na Figura 17.

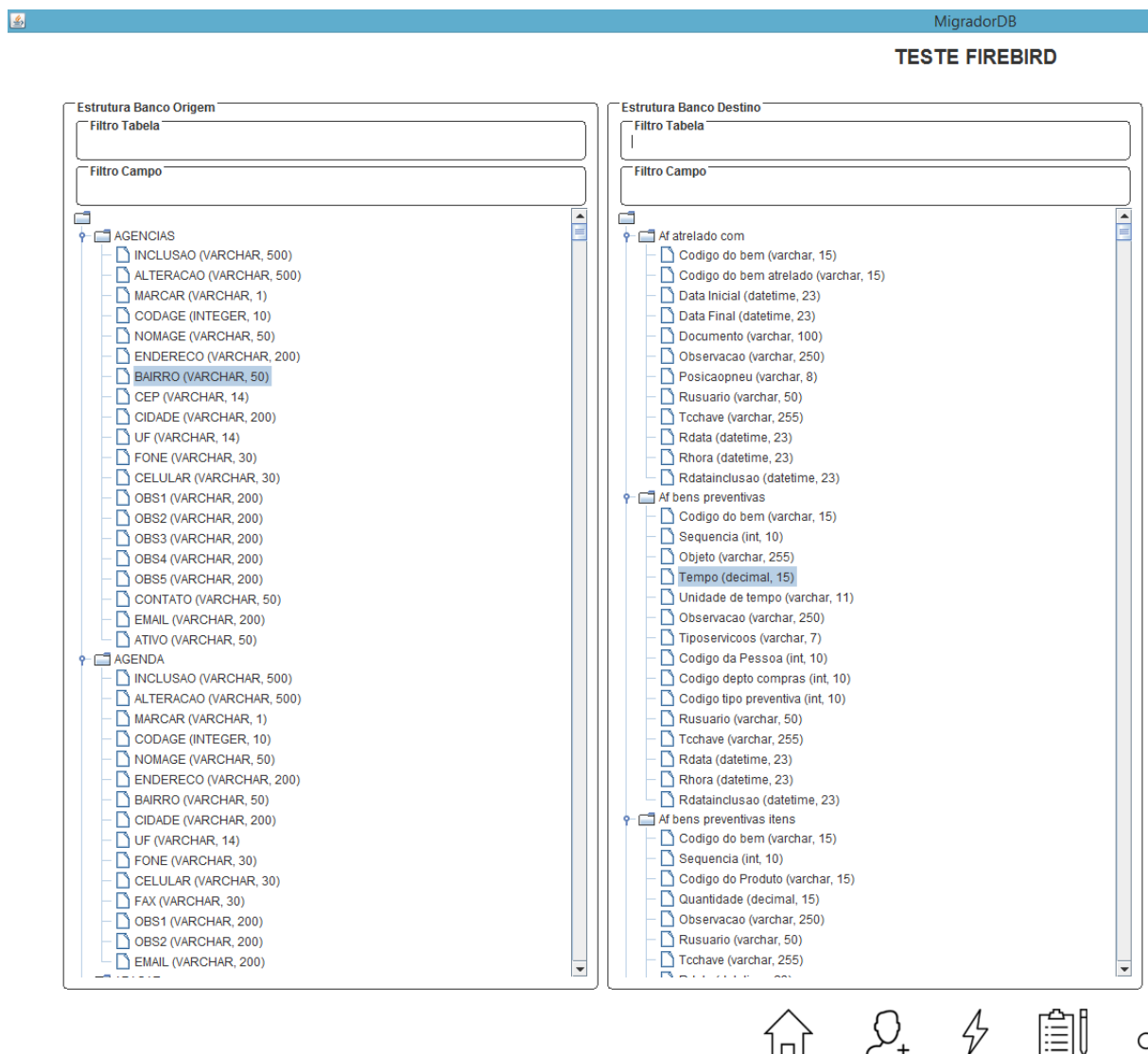
Figura 17 - Seleção do tipo de *script*



Fonte: elaborado pelo autor (2019).

Ao adicionar ou editar um *script* o sistema direcionará o usuário da ferramenta para a tela de migrações, a mesma tem o objetivo de criar e editar *scripts* e é dividida em três partes iguais, sendo que na primeira e segunda parte são exibidas a estrutura de cada banco de dados, origem e destino, suas tabelas e consultas com seus respectivos campos, tamanhos e tipos de dados. Nestas duas partes da tela, representadas na Figura 18, é possível realizar filtros pelo nome da tabela ou visões, como também filtro sobre o nome dos campos.

Figura 18 - Tela de Migração: primeira e segunda parte



Fonte: elaborado pelo autor (2019).

Portanto a Figura 18 exemplifica, ao lado esquerdo, o banco de dados de origem e seus atributos e no lado direito o banco de dados de destino e seus atributos, é nesta parte do sistema que o usuário seleciona a informação relativa ao DE (origem e vincula com o PARA (destino). Este relacionamento é feito na terceira parte da tela de *scripts*, demonstrado na Figura 19. De acordo com o tipo de *script*, são ocultadas e exibidas informações e botões específicos de cada tipo.

Figura 19 - Tela de Migração: terceira parte

Configuração

Descrição

Tipo
INSERT

Tabela/View Origem

Tabela/View Destino

SQL

Opções

Salvar Relacionar Editar Excluir Dados



Fonte: elaborado pelo autor (2019).

Para todos os tipos de *scripts* as duas primeiras partes serão sempre iguais, não há diferença de funcionalidade, diferentemente da terceira parte que sofre modificações e alterações de funcionalidades. Para os tipos de *scripts* de *create*, *delete* e *update* o painel de configuração exhibe os seguintes campos:

- Descrição: campo para o usuário preencher com a descrição do *script*;
- Tipo: conforme visto anteriormente o tipo de *script* vem com base na tela de *scripts* que é selecionado ao clicar em novo ou com base no *script* que está sendo editado;
- Tabela/View Origem: nome da tabela ou visão de origem, habilitado para edição quando o tipo de *script* for *create*;

- Tabela/View Destino: nome da tabela ou visão de origem, habilitado para edição quando o tipo de *script* for *update* ou *delete*;
- SQL: campo para escrita do comando SQL.

Para estes mesmos tipos de *scripts* (*create*, *delete* e *update*) além dos campos citados acima a tela habilita botões com funcionalidades, estas funcionalidades são:

- Salvar: valida se os campos estão preenchidos e salva o *script*;
- Testar: testar se o comando SQL escrito pelo usuário está funcionando. A ferramenta faz backup do banco de dados, testa e depois volta ao estado inicial;
- Dados: direciona o usuário para a tela de dados, que é descrita no decorrer do capítulo.

As figuras 20, 21, 22 demonstram um exemplo do painel de configuração para os tipos de *script create*, *delete* e *update* conforme citados anteriormente. As figuras demonstram os campos liberados para preenchimento como também um exemplo escrito pelo autor de cada tipo de *script*.

Figura 20 - Tela de Migração: exemplo do painel de configurações do tipo de *script create*

Configuração

Descrição

CREATE

Tipo

CREATE

Tabela/View Origem

viewPessoas

Tabela/View Destino

SQL

CREATE VIEW viewPessoas
SELECT * FROM CLIENTES
WHERE ATIVO = 0

Opções

Salvar

Testar

Dados

Fonte: elaborado pelo autor (2019).

Figura 21 - Tela de Migração: exemplo do painel de configurações do tipo de *script delete*

Configuração

Descrição
DELETE PESSOAS

Tipo
DELETE

Tabela/View Origem

Tabela/View Destino
PESSOAS

SQL
DELETE FROM PESSOAS

Opções

Salvar

Testar

Dados

Fonte: elaborado pelo autor (2019).

Figura 22 - Tela de Migração: exemplo do painel de configurações do tipo de *script update*

Configuração

Descrição

UPDATE INVENTARIO AJUSTA QTDE REF

Tipo

UPDATE

Tabela/View Origem

Tabela/View Destino

INVENTARIO

SQL

UPDATE INVENTARIO SET QUANTIDADER = QUANTIDADE
WHERE DATA = '2019-09-31'

Opções

Salvar

Testar

Dados

Fonte: elaborado pelo autor (2019).

O tipo de *script insert* tem funcionalidades diferenciadas dos demais tipos citados acima, no que diz respeito a campos a serem preenchidos. O único liberado e obrigatório é a [descrição], os demais são preenchidos e armazenados automaticamente pelas funcionalidades da tela. Neste formato de visualização, conforme Figura 23, o campo [SQL] foi substituído por uma listagem das relações já realizadas, na funcionalidade [salvar] foram acrescentadas validações e regras além de existir três novas funcionalidades, são elas:

- Relacionar: esta funcionalidade relaciona o campo da estrutura de origem com o campo da estrutura de destino. Após selecionados os campos e pressionado o botão de relacionar a ferramenta direciona o usuário para a tela de relacionamento que é explicada na sequência da seção;
- Editar: possibilita alterar o relacionamento selecionado na listagem, a ferramenta direciona o usuário a tela de relacionamento;

Excluir: exclui o relacionamento selecionado na listagem.

Figura 23 - Tela de Migração: exemplo do painel de configurações do tipo de *script insert*

Configuração

Descrição

INSERT BAIRROS

Tipo

INSERT

Tabela/View Origem

BAIRROS

Tabela/View Destino

Bairros

Relações

id	Origem	Destino	Tipo Origem
64	CODBAI	Código do Bairro	1=1
65	NOMBAI	Descricao do Bairro	1=1

Opções

Salvar

Relacionar

Editar

Excluir

Dados

Fonte: elaborado pelo autor (2019).

Conforme visto anteriormente, ao clicar em relacionar, a ferramenta direciona o usuário para a tela de relacionamento, mas antes disto é solicitado qual é o tipo de relacionamento que ele deseja criar, conforme demonstrado na Figura 24. Este tipo de relacionamento serve para que a tela de relacionamento e a rotina de migração diferencie e trate relações diferenciadas entre tabelas e campos. Os tipos de relacionamentos são:

- 1=1: relação simples de campo para campo tanto na origem quanto no destino. Como exemplo, o campo [NOMBAI] da tabela de [BAIROS] na origem seria vinculado com o campo de [descricao do bairro] da tabela de [bairros] no destino;
- N=1: relação onde o banco de origem possui uma tabela e o banco de destino possui um campo, como por exemplo, no banco de dados de origem há uma tabela de [clientes] com relacionamento em uma tabela [bairros] e no banco de destino houver somente a tabela de [pessoas] com um campo de [descricao do bairro]. O vínculo neste caso é com o campo [codigo do bairro] da origem com o campo [descricao do bairro] do destino;
- 1=N: relação onde o banco de origem possui um campo e o banco de destino possui uma tabela, neste caso ocorre o inverso do exemplo anterior, onde na origem há o campo de [descricao do bairro] e no destino há uma tabela de bairros;
- Fórmula: abre a possibilidade de o usuário escrever uma fórmula via comando SQL sobre o campo da tabela de origem. Como exemplo, caso haja um campo único na origem de CPF e CNPJ e no destino existam dois campos separados, então para este caso avalia-se o tamanho do campo da origem para atualizar o CPF ou CNPJ;
- Fixo: valor a ser inserido no campo de destino terá seu conteúdo fixo, como por exemplo, ao migrar uma tabela de origem de [fornecedor] e no destino uma tabela de [pessoas] onde esta tabela possui atributo lógico se ele é fornecedor, na migração é possível atribuir esse valor fixo.

Figura 24 - Seleção do tipo de relacionamento

usuario (varchar, 50)
 chave (varchar, 255)
 idigo do Roteiro (int, 10)
 data (datetime, 23)
 hora (datetime, 23)
 datainclusao (datetime, 23)

Seleccione uma opção

O que deseja criar?
 1=1
 1=1
 N=1
 1=N
 Fórmula
 Fixo

BAIRROS

Tabela/View Destino
 Bairros

Relações

id	Origem	
64	CODBAI	Codig
65	NOMBAI	Desc

Opções

Salvar
 Relacionar
 Editar

Fonte: elaborado pelo autor (2019).

A tela de relacionamento, apresentada na Figura 25, é composta por campos não editáveis ao usuário que são informações pertinentes ao relacionamento que está sendo criado, essas informações são: a tabela, o campo, o tipo e tamanho do campo tanto da origem quanto do destino selecionados na tela de migração.

Figura 25 - Tela de Relacionamento

MigradorDB

TESTE FIREBIRD

Nome tabela origem

BARRIOS

Nome coluna origem

NOMBAI

Tamanho coluna origem

50

Tipo coluna origem

VARCHAR

Tipo de Relacionamento

1=1

Nome

NOMBAI

Valor em Branco

Fórmula

Nome tabela destino

Barrios

Nome coluna destino

Descricao do Bairro

Tamanho coluna Destino

50

Tipo coluna Destino

varchar

Migração Chave

id	nometabela	nomecampo1	nomecampochave1	nomecampochave2






Salvar

Excluir

Carregar

Atualizar

Cancelar



Fonte: elaborado pelo autor (2019).

Os outros campos presentes na tela são exibidos e/ou bloqueados conforme o tipo de relacionamento, cada tipo de relacionamento resulta em obrigações e preenchimentos diferentes de campos, conforme demonstrado a seguir.

- 1=1: Neste tipo de relacionamento o único campo editável, mas não obrigatório é o campo [Valor em Branco] que tem o objetivo de informar um valor e caso o conteúdo do campo de origem esteja nulo, vazio ou zerado a rotina de migração substituirá por este valor fixado;
- N=1 e 1=N: Nestes tipos de relacionamento é obrigatoriamente necessário o preenchimento do painel de [Migração Chave] que é a chave estrangeira do campo com a tabela “N” de origem ou do destino. O objetivo inicial era fazer essa relação de forma automática com base na chave estrangeira gravada no banco de dados, mas devido a alguns casos de o banco de dados não possuir relacionamento entre as tabelas foi necessário a criação desta estrutura. O botão [Carregar] tenta buscar a chave estrangeira armazenado no banco de dados e caso não encontre, habilita a edição do conteúdo manualmente;
- Fórmula: Neste tipo de relacionamento o único campo editável e obrigatório é o campo “Fórmula” que tem o objetivo de permitir ao usuário

criar uma fórmula via padrão SQL para tratamento do campo na consulta das informações da origem;

- Fixo: Neste tipo de relacionamento o único campo editável e obrigatório é o campo [nome] que tem o objetivo de o usuário informar o valor fixo que será inserido no campo de destino.

Outra funcionalidade observada na tela de migração é a tela de dados, representada na Figura 26, a qual tem o objetivo de realizar consultas sobre o banco de dados de origem, no lado esquerdo da tela é possível realizar comandos SQL para que sejam exibidos na listagem do lado direito da tela. Portanto se o usuário necessitar migrar a tabela de [bairros] e deseja visualizar quais campos possuem informação ou verificar a existência de algum registro com problema é possível avaliar isso nesta tela.

Figura 26 - Tela de Dados

MigradorDB

TESTE FIREBIRD

Digite o Comando SQL:
SELECT * FROM Bairros

Executar SQL

Tabela Origem
Dados da tabela: Bairros

INCLUSAO	ALTERACAO	MARCAR	CODBAI	NOMBAI	VALTELE	CODALT
28/2011.2	28/2011.22.55.33		2	CHICO MENDES		
28/2011.2	28/2011.22.55.33		3	SALINAS		
28/2011.2	28/2011.22.55.33		4	PARQUE DOS PINOS		
28/2011.2	28/2011.22.55.33		5	GUAJUVIRAS		
28/2011.2	28/2011.22.55.33		6	COSTA DO SOL		
28/2011.2	28/2011.22.55.33		7	ZONA "A"		
28/2011.2	28/2011.22.55.33		8	TRAMANDAI SUL		
28/2011.2	Usu. NECA-CAI		1	CENTRO	20040	
28/2011.2	28/2011.22.55.33		9	NAZARE		
28/2011.2	28/2011.22.55.33		10	LOTEAMENTO MORADAS DO S.		
28/2011.2	28/2011.22.55.33		11	CAVALHADA		
28/2011.2	28/2011.22.55.33		12	NITEROI		
28/2011.2	28/2011.22.55.33		13	ESTANCIA VELHA		
28/2011.2	28/2011.22.55.33		14	SANTA CECILIA		
14/09/2011.	Usu. EVA 14/09/2		15	PARQUE DOS PINOS	18087	
22/09/2011.	Usu. ANDERSON		16	ARPOADOR	22181	
25/09/2011.	Usu. ANDERSON		17	JARDIM PIRANGA	23867	
27/09/2011.	Usu. MIRO 27/09/		18	PETROPOLIS	24926	
14/10/2011.	Usu. ANDERSON		19	PIRABEIRABA	34078	
15/10/2011.	Usu. ANDERSON		20	SANTANA	34761	
15/10/2011.	Usu. ANDERSON		21	PARTENON	34825	
17/10/2011.	Usu. ANDERSON		22	SANTANA	35514	
22/10/2011.	Usu. ANDERSON		23	SANTO ONOFRE	38826	
24/10/2011.	Usu. ANDERSON		24	MORADA DO VALE	41023	
07/11/2011.	Usu. ADRY 07/11/		25	AZENHA	50383	
09/11/2011.	Usu. ADRY 09/11/		26	RUBEM BERTA	51221	
12/11/2011.	Usu. ADRY 12/11/		27	GUAJUVIRAS	53792	
12/11/2011.	Usu. ADRY 12/11/		28	RUBEM BERTA	54267	
12/11/2011.	Usu. ADRY 12/11/		29	RO BRANCO	54441	
17/11/2011.	Usu. ADRY 17/11/		30	JARDIM ATLANTICO	58048	
19/11/2011.	Usu. ADRY 19/11/		31	BOM JESUS	60198	
22/11/2011.	Usu. ADRY 22/11/		32	DOM FELICIANO JARDIM FLO.	61852	
23/11/2011.	Usu. ADRY 23/11/		33	BOM PRINCIPIO	62945	
28/11/2011.	Usu. ADRY 28/11/		34	MATHIAS VELHO	65219	
28/11/2011.	Usu. ADRY 28/11/		35	ANTENA	65781	
28/11/2011.	Usu. ADRY 28/11/		36	XANGRILA	66298	
28/11/2011.	Usu. ADRY 28/11/		37	RAIHA DO MAR	66286	
01/12/2011.	Usu. EVA 01/12/2		38	SANTANA	68524	
02/01/2012.	Usu. ADRY 02/01/		39	SANTA RITA	94651	
03/01/2012.	Usu. ADRY 03/01/		40	GLORIA	95842	
06/01/2012.	Usu. VENDAS 06/		41	TARUBIA	109833	
25/01/2012.	Usu. EVA 25/01/2		42	PASSO D'AREIA	115567	
04/04/2012.	Usu. ANDERSON		45	GOIAS	163723	
08/06/2012.	Usu. ANDERSON		46	IPANEMA	195376	

Fonte: elaborado pelo autor (2019).

5.3.3 Rotina de exportação e importação de dados

Para um melhor entendimento de uma das principais funcionalidades da ferramenta, nesta seção será detalhada a rotina de exportação e importação de dados, que tanto para o teste quanto para a migração final se utiliza da mesma rotina. A rotina chamada de executarScript recebe por parâmetros se é teste ou migração final, o projeto e a classe de log para retornar as informações para o campo de saída da tela.

Cada linha relevante de código que a rotina executa é gerado um *log* e mostrado no campo de saída da tela de *scripts* para que o usuário saiba o que a rotina está fazendo. Este log é exibido em ordem cronológica, do mais recente para o mais novo com informações importantes para o usuário, como por exemplo, o *script* que está sendo executado, a hora que cada *script* começou e terminou, caso apresente algum erro no decorrer da rotina é demonstrado a linha do erro e as consultas que estavam sendo executadas, a quantidade de registros inseridos, entre outros.

Como citado, a rotina é utilizada tanto para o teste quanto para a migração final, na Figura 27 será demonstrada uma parte do código fonte da rotina demonstrando a separação entre esses dois casos.

Figura 27 - Código fonte exemplo de separação entre teste e migração final

```
if (bTeste) {
    boolean success = (new File(sCaminho + conexaoBancoDestino.getNomebd() + ".BAK")).delete();
    if (success) {
        logger.adicionarComHora("Backup antigo encontrado na pasta foi deletado");
    }

    BancoDestinoDAO.getInstance().backup(sCaminho);
    logger.adicionarComHora("Backup banco destino");

    BancoDestinoDAO.getInstance().restore(sCaminho);
    logger.adicionarComHora("Restauração do banco de teste");

    Class.forName(DriverBD.SQLSERVER);

    builder.append("jdbc:sqlserver://").append(conexaoBancoDestino.getInstanciadb()).append("
    conn = DriverManager.getConnection(builder.toString(), conexaoBancoDestino.getUsuariodb()

    logger.adicionarComHora("Abre conexão com banco de testes");

} else {
    conn = BancoDestinoDAO.getInstance().Conexao();
}
```

Fonte: elaborado pelo autor (2019).

Quando tipo de *script* for *create*, *update* ou *delete* a rotina basicamente executa o comando SQL escrito pelo usuário no banco de origem ou destino. Já o tipo de *script insert* exigiu uma programação maior para que o objetivo deste tipo fosse atendido. A Figura 28 representa parte deste código fonte e serve para o detalhamento da rotina após a figura.

Figura 28 - Código fonte exemplo tipo *script insert*

```

227 if (listScript.get(countScript).getTipo().equals("INSERT")) {
228     sqlInsertPadrao = montaSQLInsertOrigem(listScript.get(countScript)).replace("\\", "");
229     sqlSelectOrigem = montaSQLSelectOrigem(listScript.get(countScript));
230     List<String> listValores = new ArrayList<>();
231     List<Migracao> listCamposOrigem = montaListaCamposOrigem(listScript.get(countScript));
232     List<String> listCamposDestino = montaListaCamposDestino(listScript.get(countScript));
233     ResultSet rs = BancoOrigemDAO.getInstance().selectSQL(sqlSelectOrigem);
234     while (rs.next()) {
235         sqlSelectOrigem = sqlInsertPadrao;
236         for (int countCampos = 0; countCampos < listCamposOrigem.size(); countCampos++) {
237             switch (listCamposOrigem.get(countCampos).getTipoOrigem().toUpperCase()) {
238                 case "1=N": sValor = fMigracaoChaveIn(conn, listCamposOrigem.get(countCampos), rs.getString(listCamposOrigem.get(countCampos).getNomeCampoOrigem())); break;
239                 case "N=1": sValor = fMigracaoChaveIn(BancoOrigemDAO.getInstance().Conexao(), listCamposOrigem.get(countCampos).getNomeCampoOrigem(), rs.getString(listCamposOrigem.get(countCampos).getNomeCampoOrigem())); break;
240                 case "FIXO": sValor = listCamposOrigem.get(countCampos).getNomeCampoOrigem(); break;
241                 case "1=1": sValor = rs.getString(listCamposOrigem.get(countCampos).getNomeCampoOrigem()); break;
242                 case "FÓRMULA": sValor = rs.getString(listCamposOrigem.get(countCampos).getNomeCampoOrigem()); break;
243             }
244             if (sValor == null || sValor.equals("null") || sValor.equals("") || sValor.equals("0")) {
245                 sValor = listCamposOrigem.get(countCampos).getValorEmBranco();
246             }
247             listValores.add(sValor);
248             if (!listCamposOrigem.get(countCampos).getTipoOrigem().toUpperCase().equals("FIXO")) {
249                 sqlSelectOrigem = sqlSelectOrigem.replace("[ " + listCamposOrigem.get(countCampos).getNomeCampoOrigem() + "]", " " + sValor + " ");
250             }
251         }
252         sqlSelectOrigem = montaSQLInsertDestino(listScript.get(countScript)) + " " + sqlSelectOrigem;
253         BancoDestinoDAO.getInstance().executar(conn, trataSQL(sqlSelectOrigem));
254         cont++;
255     }
256     logger.adicionarComHora("Script " + listScript.get(countScript).getOrdem() + " inseridas " + cont + " linhas ");
257 }

```

Fonte: elaborado pelo autor (2019).

Baseado na Figura 28, a rotina inicia montando o comando SQL padrão para o *insert*, na linha 228 é chamada a rotina *montaSQLInsertOrigem* que retorna os campos da origem separados por vírgula para fazer a substituição pelos valores no decorrer da rotina. Já na linha 229 é chamada a rotina *montaSQLSelectOrigem* que retorna uma consulta sobre a tabela do banco de dados de origem com os campos que foram selecionados no *script*.

Tendo essas duas informações disponíveis, a rotina busca os dados no banco de origem conforme pode ser visto na linha 233 e percorre cada registro selecionado substituindo cada nome de campo pela informação presente neste campo no banco de dados.

Por fim a linha 252 monta o comando para inserção das informações no banco de dados e executa o mesmo na próxima linha.

5.3.4 Dificuldades encontradas

Durante o desenvolvimento da ferramenta muitas dificuldades foram encontradas, algumas foram contornadas e outras exigiram novos estudos para serem e solucionadas. A principal dificuldade encontrada foi nas estruturas dos bancos de dados, os bancos testados estavam em desacordo com as boas práticas de criação de banco de dados estudadas no curso no decorrer da graduação. Vários pontos da ferramenta tiveram que ser adaptados para contornar esses erros de estrutura, como, por exemplo, falta de relacionamento entre tabelas.

Outra dificuldade encontrada foi trabalhar com múltiplos bancos de dados com padrões SQL diferenciado, onde o desenvolvimento necessitou realizar tratamentos na montagem dos *scripts* de migrações de dados. Além disso os diferentes tipos de cada atributo necessitaram e exigiram cuidados no desenvolvimento, como por exemplo, um campo estando em padrão texto no banco de dados da origem e no destino um tipo numérico fez com que a rotina tivesse que ser tratada para validar se seriam possíveis essas migrações.

6 ANÁLISE DA FERRAMENTA

O objetivo geral da ferramenta é tornar o processo de migração de dados ágil, eficiente e eficaz diminuindo erro de importação e o tempo de implantação do novo software. Pensando nisso, no Quadro 7, são apresentados os resultados comparativos obtidos pelos três cenários a partir da documentação prévia do autor sem a utilização de ferramenta específica para migração de dados e os tempos dos mesmos cenários com a utilização da ferramenta desenvolvida neste estudo. Os tempos apresentados no Quadro 7 estão apresentados em horas.

Quadro 7 - Análise da ferramenta

Fases	Cenário A (h)		Cenário B (h)		Cenário C (h)	
	Anterior	Ferramenta	Anterior	Ferramenta	Anterior	Ferramenta
Análise e compreensão da estrutura do cliente	4	4	4	3	2	2
Análise do banco de dados	4	4	10	6	10	5
Criação do <i>Scripts</i>	35	19	33	22	41	34
Carga banco de dados	-	-	5	2	4	3
Tempo total de migração de banco de dados	43	27	52	33	57	44

Fonte: elaborado pelo autor (2019).

Para um melhor entendimento do Quadro 7 são apresentadas as ações de cada fase.

- Análise e compreensão da estrutura do cliente: é realizada a assimilação da estrutura do cliente, verificando informações pertinentes ao modelo banco de dados e sua devida conexão;
- Análise dos bancos de dados: nessa fase é identificado quais dados o cliente deseja migrar e realizada uma busca e compreensão da estrutura de tabelas e campos do cliente;
- Criação dos *scripts*: Nesta fase é realizada a montagem dos *scripts* de exportação e importação de dados, levando em consideração as avaliações dos itens anteriores;
- Carga de dados: é o processo de carga do banco de dados de origem para o de destino.

Em relação ao cenário A (PostgreSQL), neste caso foi efetuada a migração das tabelas de Pessoas, Produtos, Contas a Receber, Contas a Pagar, Notas de Saída e Notas de Entradas e seus cadastros de apoio como Bairros, Cidades, Estrutura de Estoque, entre outras. Podemos verificar que a análise, compreensão da estrutura do cliente e análise do banco de dados não houve variações dos tempos entre o processo anterior e o tempo utilizando a ferramenta, pois o cliente que estava sendo migrado conhecia a estrutura de tabelas e campos, facilitando o processo. Já no item criação dos *scripts* houve um ganho de 37% do tempo, foi o item que mais apresentou variação. Como na época que foi medido os tempos de migração não foram calculados os tempos de carga banco de dados, o mesmo também não foi feito utilizando a ferramenta.

Sobre o cenário B (Firebird), foi efetuada a migração das mesmas informações presentes no cenário A, neste caso é possível identificar que cada fase teve um ganho de tempo, representando uma redução de 36,54% no tempo total. Outro fato que vale a pena destacar, é que o cliente que utilizava essa base não tinha muitos processos de controles implantados no sistema, mas devido aos dados não estarem coerentes gerou um volume de horas mais expressivo.

Por fim em análise o cenário C (SQL Server), é possível observar que os tempos não seguiram o mesmo padrão das bases analisadas anteriormente. Nessa base é possível verificar que mesmo com a utilização da ferramenta, o tempo ganho, 22,81%, não foi tão expressivo quanto os outros dois cenários, devido ao banco de

dados de destino ser igual ao de origem a criação de *scripts* sem a utilização da ferramenta foi facilitada.

Com isso, é possível afirmar que a fase de análise e compreensão da estrutura do cliente não teve nenhuma melhora no processo com a utilização da ferramenta. No entanto, a fase de análise dos bancos de dados obteve bons resultados utilizando a ferramenta, por ser possível visualizar o banco de origem e de destino um ao lado do outro, além de poder visualizar as informações armazenadas nas tabelas de origem.

Outro item que merece destaque é criação dos *scripts*, neste item também houve redução do tempo, devido a facilidade e agilidade que a ferramenta apresentou na criação das relações entre os dois bancos de dados, com a visualização lado a lado da estrutura de origem (DE) com a estrutura do destino (PARA).

Com base nos resultados apresentados no Quadro 7, verifica-se que no processo de migração utilizando a ferramenta desenvolvida neste estudo, comparado com o cenário sem a utilização de nenhuma ferramenta, houve redução média de 32,19% no tempo total de migração de dados avaliando os três cenários (A, B e C). Ambos tiveram o mesmo comportamento de redução de tempo nas mesmas fases. Como consequência da redução do tempo de migração de dados, diminuímos o custo da operação, já que este serviço é cobrado por hora. No próximo capítulo são apresentadas as considerações finais deste trabalho.

7 CONSIDERAÇÕES FINAIS

Conforme apresentado ao longo do trabalho, as constantes mudanças impostas pelo governo e pelo mercado, fazem muitas empresas sentirem a necessidade de fazer a substituição do sistema ERP. Entretanto algumas delas acabam não efetuando a troca devido ao alto custo de implantação do novo sistema. Pensando nisso o trabalho propôs o desenvolvimento de uma ferramenta que agiliza o processo de migração entre as bases de dados fazendo com que reduza o tempo e consequentemente o custo de migração dos dados.

Os bancos de dados das empresas armazenam informações fundamentais para o funcionamento operacional da empresa como também para utilização nas tomadas de decisões do negócio, o cuidado na manipulação dessas informações é fundamental ao processo de migração de dados.

Diante do cenário apresentado, somado com a experiência do autor deste trabalho com migrações de sistemas ERP e documentações prévias realizadas pelo mesmo, foi desenvolvida a ferramenta focada na migração entre bases de dados. Ao final, foi elaborada uma tabela com a finalidade de comparar o processo anterior de migração de banco de dados com processo atual utilizando a ferramenta desenvolvida.

Levando em consideração os resultados apresentados no capítulo anterior podemos concluir que a ferramenta demonstrou ser eficiente, pois obteve um ganho de tempo considerável, principalmente nos tópicos de análise do banco de dados e montagem dos *scripts*. A mesma demonstrou ser eficaz ao reduzir erros de migração uma vez que os *scripts* estão previamente cadastrados e testado na ordem correta.

Tendo em vista que a ferramenta foi criada e pensada como um projeto apenas para este estudo, é possível afirmar que a mesma também tem potencial de se tornar uma ferramenta comercial. Como o cenário de migração de dados é amplo e espaço de tempo é curto, algumas implementações foram limitadas a atender o apenas o escopo deste trabalho. Pensando nisso, o capítulo a seguir demonstrará trabalhos futuros deste estudo.

8 TRABALHOS FUTUROS

A ferramenta desenvolvida nesse projeto foi criada e pensada para se adaptar a novos bancos de dados. Inicialmente não foi tratada como um possível produto final, a ser disponibilizada no mercado e sim apenas como um estudo de caso para este trabalho de conclusão. No decorrer do trabalho e dos testes finais a ferramenta se mostrou uma ferramenta comercial viável, com isso foram levantados alguns ajustes para que a ferramenta se torne um produto mais completo ao mercado.

A seguiram são listadas melhorias que poderão estar presentes nas próximas versões do sistema.

- Melhorias de usabilidade: para que a ferramenta não necessite de treinamentos e seja mais intuitiva é necessário realizar melhorias de usabilidade do produto;
- Padronização de comandos SQL: cada banco de dados pode variar a padronização de comandos SQL. Para que o usuário não necessite ter o conhecimento dos padrões de cada banco de dados de origem, a proposta seria a criação e utilização dos mesmos padrões do banco de dados de destino, que é o banco de dados de conhecimento do usuário, fazendo rotinas de tradução automática dos padrões;
- Complementar cadastro de projetos: criou-se uma estrutura simples para o cadastro de projetos, objetivo é poder complementar para que possamos ter mais dados do projeto armazenados, como por exemplo, o ano da migração, regime tributário do cliente, telefone, pessoa de contato, entre outras informações;

- Estrutura de destino (PARA) pronta: como o banco de dados geralmente é o mesmo e as empresas possuem campos comuns na utilização das migrações, a proposta seria deixar uma estrutura do destino (PARA) criada por padrão e utilizar as revisões em cima desse padrão;
- Armazenar log de dados: fazer com que a ferramenta armazene todas saídas de dados em um log para permitir análises futuras.

REFERÊNCIAS

ABREU, Fábio Silva Gomes da Gama e Abreu. **Desmistificando o conceito de ETL**. Disponível em: <http://www.fsma.edu.br/si/artigos/v2_artigo1.pdf>. Acessado em: 10/05/2019.

ALVES, William Pereira. **Banco de dados**. -- 1. ed. -- São Paulo : Érica, 2014. Disponível em <<https://integrada.minhabiblioteca.com.br/books/9788536518961/pageid/0>>. Acessado: 02/03/2019

AUDY, Jorge Luis Nicolas; ANDRADE, Gilberto Keller de; CIDRAL, Alexandre. **Fundamentos de sistemas de informação**. Porto Alegre: Bookman, 2011. E-book. Disponível em: <<https://integrada.minhabiblioteca.com.br/books/9788577801305/pageid/0>>. Acessado em: 08/02/2019.

BELMIRO, N. João , Belmiro N.. **Sistemas de informação**. São Paulo: Pearson Education do Brasil, 2012. E-book. Disponível em: <<http://www.univates.br/biblioteca/bibliotecavirtual-universitaria>>. Acessado em: 26/02/2019.

CAIÇARA JUNIOR, Cícero **Sistemas integrados de gestão – ERP: uma abordagem gerencial**. 2 ed. Curitiba: InterSaberes, 2015.

CARDOSO, Virgínia; CARDOSO, Giselle. **Sistema de banco de dados: uma abordagem introdutória e aplicada**. 7. ed. São Paulo: Saraiva, 2012.

CERVO, Amado Luiz; BERVIAN, Pedro Alcino; SILVA, Roberto da. **Metodologia científica**. 6. ed. São Paulo: Pearson Prentice Hall, 2007. E-book. Disponível em: <http://univates.bv3.digitalpages.com.br/users/publications/9788576050476/pages/_1>. Acessado em: 08/02/2019.

DATE, C. J.. **INTRODUÇÃO A SISTEMAS DE BANCOS DE DADOS**. 8. ed. Rio de Janeiro: Elsevier, 2003.

ELMASRI, Ramez. **Sistemas de banco de dados**. 4. ed. São Paulo: Pearson Addison Wesley, 2005.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.

GALVÃO, Michele da Costa. **Fundamentos em Segurança da Informação**. São

Paulo: Pearson Education do Brasil, 2015. Disponível em <<https://bv4.digitalpages.com.br/?term=seguran%25C3%25A7a%2520da%2520informa%25C3%25A7%25C3%25A3o&searchpage=1&filtro=todos&from=busca#/legacy/26525>>. Acessado em: 23/04/2019.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 3. ed. São Paulo: Editora Atlas S.A., 1996.

GODOI, Christiane Kleinübing; BANDEIRA-DE-MELLO, Rodrigo; SILVA, Anielson Barbosa da (Orgs). **Pesquisa qualitativa em estudos organizacionais: paradigmas, estratégias e métodos**. 2. ed. São Paulo: Saraiva, 2010.

GOMES, Cristiane Alexandra Lopes Thomas H.; VANALLE, Rosângela M.

Aspectos críticos para a implementação De sistemas erp. Piracicaba: Campus,

2001. Disponível em: <http://www.abepro.org.br/biblioteca/ENESEP2001_TR93_0166.pdf>. Acessado em: 08/03/2019.

GONÇALVES, Edson. **Dominando NetBeans**. Rio de Janeiro: Ciência Moderna, 2006.

GONZAGA, Jorge Luiz. **Dominando o PostgreSQL**. Rio de Janeiro: Moderna, 2007.

HABERKORN, Ernesto. **Um bate-papo sobre T.I.: tudo que você gostaria de saber sobre o ERP e tecnologia da informação, mas ficava encabulado de perguntar**. São Paulo: Saraiva, 2009.

HEUSER, Carlos Alberto, **Projeto de banco de dados**. 6. ed. Porto Alegre, Bookman, 2009.

HORSTMANN, Cay; CORNELL, Gary. **Core Java**. 8. ed. São Paulo: Pearson, 2010.

JANDL JUNIOR, Peter. **Java: guia do programador : atualizado para Java 7**. 2. ed. São Paulo: Novatec, 2013.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. 7. ed. São Paulo: Atlas, 2010.

LIRA FILHO, Hermann Alexandre dos Santos. **Análise comparativa das ferramentas de ETL – Kettle e Talend**. Disponível em: <<https://si.dcx.ufpb.br/wp-content/uploads/2015/12/Hermann-Alexandre-dos-Santos-Lira-Filho.pdf>>. Acessado em: 12/05/2019.

MACHADO, Felipe Nery Rodrigues, **Projeto e implementação de banco de dados**. 3. ed. São Paulo : Érica, 2014.

MENDONÇA, Manoel Heleno Ramos de . **Metodologia de migração de dados em um contexto de migração de sistemas legados**.:UFPEL, 2009. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/1934>>. Acessado: 02/02/2019.

MEDEIROS, Luciano Frontino de. **Banco de dados principios e praticas** . Curitiba: InterSaberes, 2013. E-book. Disponível em: <<https://bv4.digitalpages.com.br/?filtro=livros&term=BANCO%2520DE%2520DADOS&searchpage=1&from=busca&page=-3§ion=0#/legacy/6289>>. Acessado em: 28/02/2019.

MILANI, André. **PostgreSQL: guia do programador**. São Paulo: Novatec, 2008

OLIVEIRA, C. S; MARCELINO M. A. **Metodologias e estratégias de migração de dados**. V. 13, n.3, p. 183-191. São Paulo: 2012. Disponível em <<https://ojs.ifsp.edu.br/index.php/sinergia/issue/view/19/38>>. Acessado em: 20/04/2019.

PFLEEGER, Shari Lawrence; FRANKLIN, Dino. **Engenharia de software: teoria e prática**. 2. ed. São Paulo: Prentice Hall, 2004.

RAMARKRISHNAN, Raghu. **Sistemas de gerenciamento de banco de dados**. 6. ed. Porto Alegre: Bookman, 2012.

RIBEIRO, Alexandre Lima. **Processos de implantação e migração de dados com utilização de ETL para um ERP comercial**. Disponível em: <http://www.ulbra.inf.br/joomla/images/documentos/TCCs/2010_2/tcc2-si-alexandre_lima_ribeiro.pdf>. Acessado em: 12/05/2019.

ROESCH, Sylvia Maria Azevedo. **Projetos de estágio e de pesquisa em administração**: guia para estágios, trabalhos de conclusão, dissertações e estudos de caso. 3. ed. São Paulo: Atlas, 2013.

ROMAN, Anderson Vidart, NOTARI, Daniel Luis . **Implantação de um Sistema Arquivístico Unificado**.: UCS, 2018. Disponível em: <<https://repositorio.ucs.br/xmlui/bitstream/handle/11338/3929/TCC%20Andersom%20Vidart%20Roman.pdf?sequence=1&isAllowed=y>>. Acessado em: 13/04/2019.

ROSINI, Alessandro Marco; PALMISANO, Angelo. **Administração de sistemas de informação e a gestão do conhecimento**. 2. ed.. São Paulo: Cengage Learning, 2012.

SANTOS, Aldemar de Araujo. **Informática na empresa**. 3. ed. São Paulo: Atlas, 2003.

SANTOS, Aldemar de Araujo. **Informática na empresa**. 5. ed. São Paulo: Atlas, 2009.

SANTOS, Pedro de Alcântara dos; RODRIGUES, Josine. **Requisitos para ferramentas de migração de dados**. Piauí, 2013. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2013/0079.pdf>>. Acessado em: 11/09/2019.

SELZLEIN, Willian Ivo. **Migração e testes de um ERP do SGDB SQL Server para o SGDB PostgreSQL**. Rio Grande do Sul, 2009. Disponível em: <<https://repositorio.ucs.br/xmlui/bitstream/handle/11338/1209/TCC%20Willian%20Ivo%20Selzlein.pdf?sequence=1&isAllowed=y>>. Acessado em: 20/08/2019.

SÊMOLA, M. **Gestão da segurança da informação: visão executiva da segurança da informação**. Rio de Janeiro: Elsevier, 2003.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S.. **SISTEMA DE BANCO DE DADOS**. 3. ed. São Paulo: Makron Books, 1999.

SOMMERVILLE, Ian; MELNIKOFF, Selma Shin Shimizu; ARAKAKI, Reginaldo. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SOUZA Cesar Alexandre de; ZWICKER, Ronaldo. **Ciclo De Vida De Sistemas ERP**. São Paulo USP, 2000. Disponível em: <http://www.interuni.com.br/academia/computacao/erp_cs.pdf>. Acessado em: 11/05/2019.

STAIR, Ralph M, REYNOLDS, George W. **Princípios de sistemas de informação Banco de dados**. São Paulo: Cengage Learning, 2015.

VERGARA, Sylvia Constant. **Projetos e Relatórios de Pesquisa em Administração**. P. 7. ed. São Paulo: Atlas, 2006.

VICCI, Claudia. **Banco de dados**. 1. ed. São Paulo: Pearson Education do Brasil, 2014.