



Jobson Rocha Pereira

Ferramenta para otimizar a importação dos dados de novos clientes do sistema Flowup

Recife

2021

Jobson Rocha Pereira

Ferramenta para otimizar a importação dos dados de novos clientes do sistema Flowup

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Cleviton Vinicius Fonsêca Monteiro

Recife

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- P436f Pereira, Jobson Rocha
 Ferramenta para otimizar a importação dos dados de novos clientes do sistema Flowup / Jobson Rocha
Pereira. - 2021.
 88 f. : il.
- Orientador: Cleviton Monteiro.
 Inclui referências e apêndice(s).
- Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco,
Bacharelado em Sistemas da Informação, Recife, 2021.
1. Migração de Dados. 2. Cloud ERP. 3. ETL. 4. Design Thinking. I. Monteiro, Cleviton, orient. II. Título

CDD 004

JOBSON ROCHA PEREIRA

Ferramenta para otimizar a importação dos dados de novos clientes do sistema Flowup

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 03 de Março de 2021.

BANCA EXAMINADORA

Cleviton Vinícius Fonsêca Monteiro (Orientador)

Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Gabriel Alves de Albuquerque Júnior

Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

“Dedico este trabalho à minha esposa e companheira de todas as horas, que sempre me incentivou. E também aos meus pais, que estiveram ao meu lado em todos os momentos da minha vida”

Agradecimentos

Agradeço a Deus por me dar a força necessária para conciliar emprego, estágio e o desenvolvimento deste trabalho.

Agradeço a minha família pelo apoio de sempre e especialmente aos meus pais pelo carinho, pela atenção, pelos ensinamentos de vida, pela dedicação, pelo cuidado e pela luta, com o intuito de disponibilizar para mim e meus irmãos uma educação de qualidade. O que sou hoje eu devo a eles.

À Renata, minha esposa, por estar ao meu lado em todos os momentos. Agradeço pelo carinho, pelo cuidado, pelo companheirismo, pela troca de conhecimentos, pela compreensão e pela dedicação.

À professora Juliana, ex-diretora da UAEADTec e ao professor Jorge, diretor atual, por me apoiarem a continuar investindo nos estudos. Agradeço ainda ao professor Jorge pelas dicas na escrita deste trabalho.

Agradeço também ao meu orientador por me mostrar os melhores caminhos para o desenvolvimento deste trabalho.

A toda equipe do Flowup por colaborar nas entrevistas e reuniões, como também, pelo apoio técnico no desenvolvimento, sem isso, não seria possível realizar este trabalho.

Aos meus colegas do curso, especialmente para Lucas Vieira e Wellington Antônio, pela colaboração e compartilhamento de conhecimento em várias disciplinas, principalmente nos trabalhos em equipe. Juntos conseguimos avançar e evoluir, tanto profissionalmente quanto como pessoa.

Agradeço aos professores do curso de Sistemas de Informação da UFRPE, especialmente aos meus mestres, Assad, Ceça, Cleviton (orientador), Gabriel, Glauco, Jones, Roberta, Rodrigo Soares e Victor, pela dedicação e ensinamentos durante o curso, e por provocar e instigar a nós alunos com o objetivo de nos fazer acreditar que podemos fazer sempre melhor.

E por último e não menos importante agradeço à coordenação do curso de Sistemas de Informação da UFRPE por atender nossas solicitações em benefício de todos os alunos e professores do curso.

*“O insucesso é apenas uma oportunidade para recomeçar de novo com mais
inteligência.”
(Henry Ford)*

Resumo

A migração de dados entre sistemas ainda é uma missão complexa para os desenvolvedores de software atualmente. Isso tem uma considerável contribuição dos vários tipos de bancos de dados que existem e a falta de padrão na construção dos projetos. Esta mesma dificuldade é percebida para os sistemas de gestão empresarial online, conhecidos como Cloud ERP (*Cloud Enterprise Resource Planning*). Estes sistemas online podem integrar todas as áreas da empresa em uma única plataforma. Em caso de haver algum problema na distribuição deste serviço online, pode ser gerado um grande prejuízo à empresa contratante, que fica totalmente dependente do sistema contratado e da empresa fornecedora. Em alguns casos, este problema pode ser resolvido com a troca do sistema *Cloud ERP*. Devido a esta demanda, foi desenvolvido o Migrup, ferramenta proposta neste trabalho, que tem a finalidade de facilitar, agilizar e flexibilizar a importação dos dados de uma planilha para um sistema *Cloud ERP*, o Flowup. Dessa forma, a empresa contratante de um outro sistema poderá exportar os dados de seu sistema para uma planilha; com isso, a empresa terá a possibilidade de migrar para o Flowup usando o Migrup para importar seus dados. O Migrup foi desenvolvido usando a metodologia do *Design Thinking* principalmente para tentar construir uma solução criativa e, assim, atender às necessidades do usuário. Em seguida, na etapa da implantação, foi definido que seria utilizado o processo ETL (*Extract Transform Load*), que é muito utilizado na migração de dados entre sistemas, com o objetivo de extrair os dados de uma fonte, ajustar e adequar estes dados e, depois, carregá-los em um destino. Dessa forma, foi usada uma ferramenta ETL para facilitar a implementação. O uso dessa ferramenta, através de um ambiente gráfico, com várias opções de componentes com finalidades diferentes, facilita na implementação e torna a manutenção mais rápida e prática. Comparado às ferramentas de importação dos outros sistemas citados neste trabalho, incluindo a ferramenta legada do sistema Flowup, o Migrup engloba quase todas as funcionalidades e características presentes nestas outras ferramentas, trazendo vários benefícios aos usuários em uma única ferramenta. Os principais foram: permitir que o próprio cliente use a ferramenta, importar de qualquer planilha com formato de tabela, agilizar a detecção de erros, corrigir o formato de alguns dados e apresentar o relatório da execução do processo.

Palavras-chave: Migração de Dados, Cloud ERP, ETL, Design Thinking.

Abstract

The data migration between systems is still a complex task for software developers today. This has a considerable contribution from the various types of databases that exist and the lack of standard in the construction of projects. This same difficulty is perceived for online business management systems, known as Cloud ERP (Cloud Enterprise Resource Planning). These online systems can integrate all areas of the company on a single platform. And if there is a problem, be it a failure in some functionality, inconsistent data, or even a failure in the connection with the server, a great loss can be generated to the company that hired the system. In these cases, the contracting company is totally dependent on the system with its supplier company. In this context, the problem can be solved by changing the Cloud ERP system. With the aim to facilitate, streamlining and making flexible the import of data from a spreadsheet into a Cloud ERP system, it is proposed in this work the Migrup tool. In this way, the contracting company of another system will be able to export the data from its system to a spreadsheet; with that, the company will have the possibility to migrate to Flowup using Migrup to import its data. Migrup was developed using the Design Thinking methodology to build a creative solution and, thus, meet the user's needs. Then, in the implementation stage, it was defined that the ETL (Extract Transform Load) process would be used, which is widely used in the migration of data between systems, with the objective of extracting data from a source, adjusting and adapting this data and , then load them to a destination. Thus, an ETL tool was used to facilitate the implementation. The use of this tool, through a graphical environment, with several options of components for different purposes, facilitates the implementation and makes maintenance faster and more practical. Compared to the import tools of the other systems mentioned in this work, including the legacy tool of the Flowup system, Migrup encompasses almost all the features and characteristics present in these other tools, bringing several benefits to users in a single tool. The main ones were: available to customers, import any spreadsheet format, faster error detection, correction in the format of some data and presentation of a report after the import process execution.

Keywords: Data Migration, Cloud ERP, ETL, Design Thinking.

Lista de ilustrações

Figura 1 – Fluxograma do processo de importação usando a ferramenta legada.	17
Figura 2 – Estrutura geral para processos ETL.	21
Figura 3 – Exemplo de um Fluxo de Controle e alguns fluxos de dados adicionados	24
Figura 4 – Exemplo do ambiente de implementação/configuração dos componentes de um Fluxo de Dados. Fonte: Elaborado pelo autor.	25
Figura 5 – Exemplo de configuração do componente <i>Derived Column</i>	25
Figura 6 – Configuração da tarefa <i>Execute SQL</i>	26
Figura 7 – Exemplo de configuração do Componente <i>Lookup</i>	29
Figura 8 – Criação de variáveis no pacote SSIS	31
Figura 9 – Criação de parâmetros no pacote SSIS	32
Figura 10 – Arquitetura da base de dados do Catálogo SSIS	32
Figura 11 – <i>DataFlow</i> na Execução do pacote ETL no modo de depuração (<i>debugging</i>)	33
Figura 12 – Código de execução padrão gerado na configuração do pacote ETL	34
Figura 13 – Código adicional para execução do pacote ETL com parâmetros	34
Figura 14 – Processo ELT típico	35
Figura 15 – Fluxo da ferramenta de importação atual do Flowup	36
Figura 16 – Fluxo das etapas utilizadas no desenvolvimento do Migrup	37
Figura 17 – Cópia da tabela do Excel para importação do sistema Nibo	49
Figura 18 – Mapeamento das colunas com o campo da entidade Fornecedores do sistema Nibo. Fonte: Elaborado pelo autor.	50
Figura 19 – Tela inicial do protótipo para importar a planilha.	53
Figura 20 – Tela de mapeamento entre as bases.	53
Figura 21 – Visualização de uma aba da planilha	54
Figura 22 – Relatório do processo de migração dos dados	54
Figura 23 – Tabela de erros encontrados na planilha	55
Figura 24 – Fluxograma do processo de importação dos dados usando o Migrup.	56
Figura 25 – Arquitetura do Migrup	58
Figura 26 – Painel para escolher o arquivo e importar a(s) planilha(s).	60
Figura 27 – Painel de mapeamento mostrando a tabela <i>Conta_Bancaria</i>	61
Figura 28 – Exemplo de dados invalidados apresentados na tabela de erros	62
Figura 29 – Experimento 1: Gráficos apresentados no relatório da migração	63
Figura 30 – Experimento 1: Gráficos apresentados no relatório da migração	63
Figura 31 – Experimento 2: Gráficos apresentados no relatório da migração	64
Figura 32 – Experimento 2: Gráficos apresentados no relatório da migração	64

Figura 33 – Mensagem informando que o arquivo importado não foi validado . . .	73
Figura 34 – Código para coleta dos metadados do arquivo Excel	73
Figura 35 – <i>script</i> para coleta dos metadados da base MAPDB	74
Figura 36 – Fluxograma geral da implementação das validações, correspondendo a cada dado importado. Fonte: Elaborado pelo autor.	75
Figura 37 – Chaves estrangeiras em alguns campos da base Cash, à esquerda, representadas pelos campos correspondentes da tabela no MAPDB, à direita.	75
Figura 38 – Configuração das expressões do <i>DataFlow Task</i> da entidade Cliente	77
Figura 39 – <i>DataFlow Task</i> com os componentes da entidade Cliente.	77
Figura 40 – Implementação da contagem dos campos nulos no pacote ETL . . .	78
Figura 41 – Configuração do componente Split na contagem de um campo Nulo	78
Figura 42 – Configuração da tarefa do tipo <i>Execute SQL Task</i>	79
Figura 43 – <i>script</i> adicionado no <i>Execute SQL Task</i> para inserção dos valores das variáveis na contagem dos campos nulos. Fonte: Elaborado pelo autor.	79
Figura 44 – <i>ControlFlow</i> da migração implementada no ambiente de desenvolvimento do SSIS.	80
Figura 45 – Configuração da conexão com a base de origem, MAPDB	80
Figura 46 – Configuração do componente Derived Column	81
Figura 47 – Configuração do componente Data Conversion	82
Figura 48 – Configuração do componente <i>Lookup</i> para pegar o identificador do fornecedor na tabela <i>Suppliers</i> no Cash. Fonte: Elaborado pelo autor.	82
Figura 49 – Configuração do componente <i>OLE DB Command</i>	83
Figura 50 – Configuração da conexão com a base de destino, Cash	83
Figura 51 – Mapeamento dos campos da base de origem (MAPDB), já transformados, com a base de destino (Cash) da entidade Fornecedor. Fonte: Elaborado pelo autor.	84
Figura 52 – Configuração do pacote ETL, disponível após o <i>Deploy</i>	85
Figura 53 – Comparação entre os campos de uma tabela da base de destino (à esquerda) e os campos de sua tabela correspondente na base intermediária (à direita).	86

Lista de tabelas

Tabela 1 – Tabela AS-IS X TO-BE da ferramenta de importação dos dados dos clientes do sistema Flowup. Fonte: Elaborado pelo autor.	50
Tabela 2 – Comparativo das ferramentas de importação dos dados de clientes, fornecidas pelos sistemas de gestão citados neste trabalho, com o Migrup.	65

Lista de abreviaturas e siglas

ASP	Active Server Pages
DT	Design Thinking
ETL	Extract Transform Load
ERP	Enterprise Resource Planning
IDE	Integrated Development Environment
IIS	Internet Information Services
MRP	Material Requirement Planning
MRP II	Manufacturing Resources Planning
MS	Microsoft
MVC	Model-View-Controller
OLE DB	Object Linking and Embedding Database
SaaS	Software as a Service
SGBD	Sistema de Gerenciamento de Banco de Dados
SIGE	Sistema Integrado de Gestão Empresarial
SQL	Standard Query Language
SSIS	SQL Server Integration Services
SSMS	SQL Server Management Studio
T-SQL	Transact-SQL
UX	User Experience
UI	User Interface
VS	Visual Studio (Microsoft)
VB	Visual Basic
XML	Extensible Markup Language

Sumário

	Lista de ilustrações	7
1	INTRODUÇÃO	13
1.1	Motivação e Justificativa	14
1.2	Objetivos	15
1.3	Organização do trabalho	15
2	REFERENCIAL TEÓRICO	16
2.1	Flowup - Sistema de Gestão Online	16
2.2	Design Thinking - DT	17
2.2.1	Perfil de Personalidade de um <i>Design Thinker</i>	18
2.2.2	Etapas do Design Thinking	19
2.3	Extração, Transformação e Carga de dados - ETL	20
2.3.1	Etapas do Processo ETL	21
2.3.2	Ferramentas ETL	22
2.3.3	SQL Server Integration Services - SSIS	23
2.3.4	Objetos Dinâmicos no Pacote SSIS	30
2.4	Solução alternativa	34
2.4.1	<i>Extract Load Transform (ELT)</i>	34
3	MÉTODO	36
3.1	<i>Design Thinking - DT</i>	37
3.1.1	Imersão	37
3.1.2	Análise e Síntese	39
3.1.3	Ideação	39
3.1.4	Prototipação	39
3.2	Desenvolvimento	40
3.2.1	Coleta dos Metadados	41
3.2.2	Mapeamento entre as Tabelas	41
3.2.3	Migração dos Dados	43
3.2.4	Relatório dos Resultados	45
3.3	Ferramentas Utilizadas	45
3.3.1	Desenvolvimento de Ferramentas Auxiliares	46
4	RESULTADOS DO PROCESSO DESIGN THINKING	48
4.1	Imersão	48

4.2	Análise e Síntese	50
4.3	Ideação	51
4.4	Prototipação	52
5	MIGRUP	56
5.1	Descrição do processo de importação usando o Migrup	56
5.2	Arquitetura do Migrup	58
5.3	Uso do Migrup	59
5.3.1	Coleta dos metadados	59
5.3.2	Mapeamento entre as tabelas	60
5.3.3	Relatório dos resultados	62
5.4	Discussão	64
6	CONCLUSÃO	67
	REFERÊNCIAS	69
A	APÊNDICE	72
A.1	Ferramentas utilizadas na implementação	72
A.1.1	MS Visual Studio (VS)	72
A.1.2	MS SQL Server Integration Services (SSIS)	72
A.1.3	MS SQL Server Management Studio (SSMS)	72
A.2	Detalhes de implementação do Migrup	72
A.2.1	Coleta dos metadados	73
A.2.2	Mapeamento entre as tabelas	74
A.2.3	Implementação do pacote ETL	76
A.2.4	<i>Deploy</i> do pacote ETL	84
A.2.5	Execução do Pacote ETL	84
A.2.6	Relatório dos resultados	85
A.3	Implementação das ferramentas auxiliares	85
A.3.1	Criação de uma base intermediária	86

1 Introdução

A cada dia novas tecnologias digitais vão surgindo. Outras são atualizadas e/ou melhoradas. A quantidade de pessoas ou empresas que utilizam essas tecnologias só tende a crescer e isso gera uma maior necessidade de evolução para as empresas que desenvolvem essas soluções tecnológicas.

O Flowup é um *Sistema Integrado de Gestão Empresarial (SIGE)*. Este tipo de software é mais conhecido como Sistema ERP (*Enterprise Resource Planning*). Há outros softwares como o Flowup atualmente no mercado e funcionam totalmente online, e por isso são chamados de *Cloud ERP* (CHEN, 2015).

Esta realidade atual é imensamente diferente de como ocorria para as empresas no passado. A gestão das empresas com ajuda da tecnologia foi iniciada no final da década de 1950, com o surgimento de novos conceitos de controle e gestão corporativa (FERRO; FERREIRA, 1999). Na década de 1960, os primeiros sistemas de gestão foram implementados nos primeiros computadores de grande porte, chamados de *mainframes*, que a princípio tinham a capacidade de controlar os estoques. Daí em diante os *sistemas de gestão* evoluíram, ao mesmo tempo que os computadores, para sistemas MRP (*Material Requirement Planning*) nos anos 1970, e MRP II (*Manufacturing Resources Planning*) na década de 1980 (SILVA; PESSOA, 2004).

A partir dos anos 1990, os primeiros sistemas de gestão começaram a integrar as várias áreas de uma empresa em um único sistema, e assim puderam ser chamados de ERP. Nasceram assim os sistemas para gerenciar as atividades, produtividade e finanças das empresas, e ainda disponibilizar todos os dados de forma integrada aos gestores. À princípio poderiam ser implantados localmente nos computadores das empresas ou em uma rede de computadores.

Houve então o surgimento da Internet, ainda na década de 1990, mas só a partir de 2012, quando a computação em nuvem (*cloud computing*) se firmou e ficou mais acessível, os sistemas ERP, assim como outros sistemas, passaram a ser disponibilizados de forma online, chamados de *Cloud ERP*, que pertencem à categoria de *Software as a Service* (SaaS) entregando aplicativos pela Internet a vários usuários finais e empresas (ELMONEM, 2016).

As vantagens dos sistemas ERP, de forma geral, são principalmente a redução de custos, aumento da produtividade e do fluxo interno de informação, colaborando assim para a integração entre os departamentos. Além disso, dão facilidade e agilidade na obtenção de informações do negócio. Desta forma, otimizam o processo de tomada de decisões das empresas. Somado a isso, são inúmeros os benefícios de um

sistema *Cloud ERP*: custos iniciais e operacionais mais baixos, rápida implantação, atualizações e upgrades rápidos e a opção de experimentar o sistema de forma grátis (ELMONEM, 2016).

As principais desvantagens estão relacionadas ao custo geral da implantação e se vale a pena o investimento, a resistência ao uso da nova ferramenta por parte dos funcionários mais conservadores, e ainda, a dependência da empresa de tecnologia que implantou o sistema (FERRO; FERREIRA, 1999). No caso desta última, todas as atividades e processos da empresa dependem do sistema, e pode ocorrer da empresa não estar mais satisfeita com os recursos oferecidos, então a empresa teria que contratar um sistema de outro fornecedor. Porém, a migração para um outro sistema não é tão simples, isso, pelo fato de ainda não haver mobilidade nesses tipos de serviços, e assim a empresa fica presa ao sistema legado.

Os sistemas ERP devem acompanhar a evolução do mercado como também as mudanças legais tanto no Brasil quanto no mundo, assim como respeitar as particularidades das empresas. Então a alta gestão da empresa deve avaliar de forma estratégica se seu sistema ERP atual ainda está adequado a suas necessidades, e caso decida trocar de sistema, que seja feita de forma planejada (COLONETTI, 2017).

Uma das principais preocupações na hora de trocar de sistema deve ser a migração dos dados. Os sistemas legados guardam todos os dados históricos da empresa, desde sua implantação. Estes dados são vitais para a empresa. Todas as atividades, processos e decisões dependem da integridade dos dados registrados pelo sistema ERP legado (OLIVEIRA; MARCELINO, 2012). Então é desejado que o novo sistema ofereça o recurso para importar esses dados.

1.1 Motivação e Justificativa

Atualmente o Flowup possui uma ferramenta para importar uma planilha eletrônica pré-formatada com os dados de novos clientes. Essa planilha é enviada ao novo cliente, que a preenche manualmente e devolve. Essa funcionalidade fica disponível apenas para a equipe técnica, que então recebe a planilha preenchida e realiza a importação para o sistema do Flowup. Neste processo atual de importação, todos os dados devem ser preenchidos conforme padrão da planilha pré-formatada de forma manual, demandando assim muito tempo. Além disso, erros no preenchimento com relação ao formato dos dados são muito comuns.

O foco deste trabalho é desenvolver uma ferramenta que otimize a importação dos dados de novos clientes ao sistema Flowup. Durante o processo de importação, esta ferramenta poderá oferecer alguns benefícios como, disponibilizar a importação dos dados de qualquer formato de planilha, realizar a validação e limpeza dos dados

antes de carregá-los no sistema Flowup e apresentar um relatório informando o resultado do processo.

A ferramenta proposta será importante tanto para a empresa desenvolvedora do Flowup como para as empresas que precisarem trocar seu sistema *Cloud ERP*. Isso, primeiramente, por reduzir a barreira de entrada de novos clientes, e assim atrair mais clientes. E depois, com a implementação das melhorias propostas, o processo ganhará em agilidade, flexibilidade e precisão, evitando retrabalho e reduzindo custos.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver uma ferramenta de importação de dados flexível e configurável que agilize e facilite o processo de importação das informações do novo cliente e integrá-la ao Flowup.

Os objetivos específicos para a construção da ferramenta são:

- Coletar os metadados da base de destino e as informações da planilha;
- Criar uma interface gráfica para a configuração e mapeamento da importação;
- Mapear os campos das duas bases;
- Utilizar um processo eficiente para extrair, limpar e transformar os dados da base de origem; e, na sequência, carregá-los na base de destino.

1.3 Organização do trabalho

O presente trabalho está organizado em seis capítulos dos quais o primeiro é a introdução, e na sequência, os outros cinco estão descritos da seguinte forma: no Capítulo 2 é apresentado o referencial teórico com os fundamentos e descrição das ferramentas necessárias para o desenvolvimento deste trabalho; no Capítulo 3 são apresentados os métodos, ferramentas, dados e materiais utilizados no trabalho; no Capítulo 4 são apresentados os resultados do processo de *Design Thinking* utilizado como metodologia no desenvolvimento; no Capítulo 5 são apresentados os resultados do desenvolvimento da ferramenta proposta neste trabalho, nomeada de Migrup, no sistema Flowup; e no Capítulo 6 são descritas as conclusões tiradas a partir dos resultados alcançados e os trabalhos futuros relacionados a este contexto.

2 Referencial Teórico

Este capítulo descreve as ferramentas e sistemas usados como base para implementação da solução proposta neste trabalho, Migrup.

As principais funcionalidades do sistema de gestão Flowup são descritas na Seção 2.1, incluindo a ferramenta legada de importação dos dados de clientes. A Seção 2.2 descreve os conceitos básicos e as etapas mais usadas do *Design Thinking*. Na Seção 2.3 são apresentados os fundamentos do processo de extração, transformação e carga dos dados, conhecido como ETL (*Extract, Transform, Load*) e a ferramenta que será utilizada para aplicar este processo. Por fim, na Seção 2.4, são descritos os fundamentos de uma ferramenta alternativa para o desenvolvimento do trabalho.

2.1 Flowup - Sistema de Gestão Online

O Flowup é um sistema *Cloud ERP* para empresas, que permite o gerenciamento de projetos, equipes e finanças das empresas (FAST, 2021). Suas principais funcionalidades estão descritas nas subseções abaixo. No Item 2.1 é apresentado o processo de importação da ferramenta legada.

Gestão Financeira

O sistema permite o controle financeiro de forma online, disponibilizando fluxo de caixa, receitas e despesas, orçamentos, relatórios financeiros, dentre outros. É possível verificar os valores já compensados, as contas a pagar e receber, a evolução das despesas mensais, saldo das contas bancárias, além de outras informações, no *dashboard*. Que segundo Few e Edge (2007) é um painel visual com as informações mais importante da empresa em uma única tela. Dessa forma, estas informações podem ser monitoradas de forma rápida pelo. O Flowup oferece também a importação de extratos bancários diretos para o sistema, entre outras funcionalidades associadas à gestão das finanças de uma empresa.

Controle de Horas

Nesta área o Flowup possibilita as seguintes funcionalidades: controle do banco de horas dos funcionários; relatórios financeiros com horas trabalhadas por projeto e seus custos; configuração do rateio de custos para geração automática de relatórios; dentre outras.

Gestão de Projetos

As principais funcionalidades na área de projetos são: quadro kanban com cards de tarefas, onde cada quadro pertence a um projeto; relatórios de tarefas configurável, onde é possível filtrar por período, projeto, podendo refinar a busca por prazo, responsável ou pelo status da tarefa.

Ferramenta legada de importação de dados de novos clientes

Esta ferramenta funciona em um ambiente separado do Flowup, chamado de ControlPanel, onde apenas a equipe do Flowup tem acesso. Assim, o técnico responsável pela importação dos dados deve enviar uma planilha, pré-formatada, ao cliente que solicitou esta ação. Esta planilha, é preenchida e devolvida ao técnico, que realiza a importação dos dados desta ao sistema Flowup. Pode ser verificado na Figura 1, o fluxograma da execução do processo de importação usando a ferramenta legada.

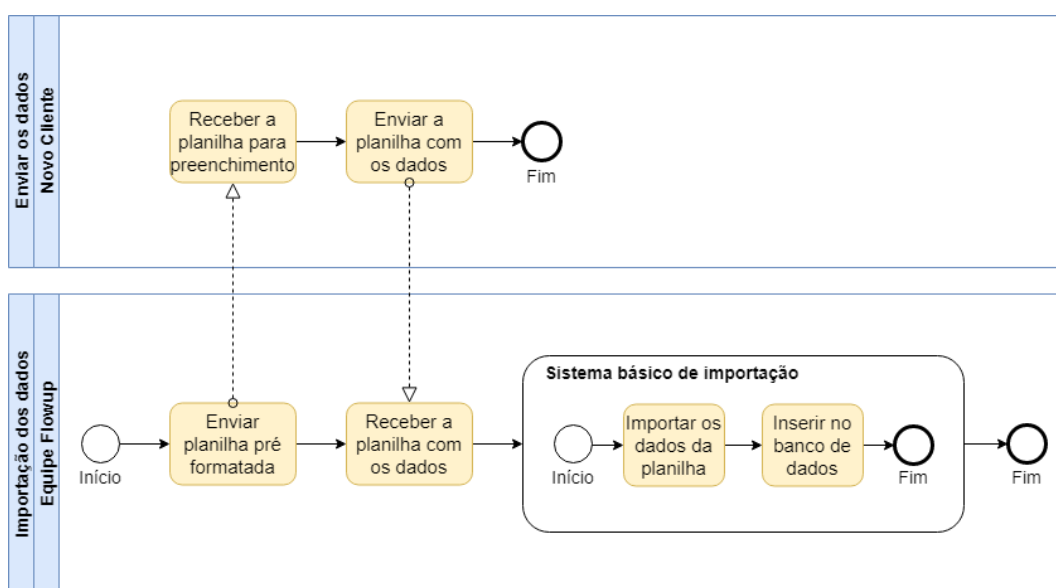


Figura 1 – Fluxograma do processo de importação usando a ferramenta legada.

Fonte: Elaborado pelo autor.

2.2 Design Thinking - DT

O Design Thinking (DT) surgiu da busca das empresas por novos caminhos de inovar. Para as empresas, inovar era criar novas soluções tecnológicas, porém, na década de 1990, uma nova filosofia criada por vários consultores empresariais dos Estados Unidos, estava sendo adotada, chamada de *Total Quality Management*, que tem o objetivo de agregar qualidade aos produtos e processos de uma organização (VIANNA, 2012).

Depois, este controle de qualidade também passou a ser baseado na satisfação dos clientes, processo este crucial para a sobrevivência das empresas a partir dessa época até os dias atuais. Esta ideia foi desenvolvendo-se até se transformar no DT, que é definida por [Vianna \(2012\)](#) como uma abordagem focada no ser humano, capaz de produzir soluções inovadoras para negócios.

De acordo com [Megido \(2017\)](#) esta abordagem possui três pilares fundamentais: a empatia, que ajuda na ampliação da visão do problema analisando as diferentes perspectivas através da multidisciplinaridade; a colaboração, que usa as ideias de pessoas diferentes para serem trabalhadas juntas com o objetivo de inovar; e experimentação, que absorve as informações das tentativas que não deram certo para realimentar o processo, e assim conseguir soluções melhores.

Esse conjunto multidisciplinar permite que um problema seja analisado por diversas perspectivas conforme a especialidade de cada membro da equipe, cada um com sua visão. Isto é apoiado pelo trabalho colaborativo, aumentando assim as chances de resultar na criação de soluções mais criativas.

Para [Pinheiro e Alt \(2018\)](#), o DT não é uma moda ou tendência, trata-se de um modelo mental, é a atitude dos profissionais e organizações de criar produtos que atendam às necessidades das pessoas ou empresas através da sensibilidade e os métodos do Design.

2.2.1 Perfil de Personalidade de um *Design Thinker*

[Vianna \(2012\)](#) define *Design Thinkers* como pessoas que desafiam os padrões de pensamento, comportamento e de sentimento. Dessa forma são pessoas capazes de criar soluções que expandem a percepção humana, gerando novos significados ao produto desenvolvido e estimulando os aspectos cognitivo, emocional e sensorial de seus usuários.

Na opinião de [Brown \(2008\)](#), muitas pessoas que não trabalham diretamente como designer profissionalmente podem, de forma natural, já possuírem o pensamento de um. Ele afirma que se uma pessoa passar pelas experiências corretas durante seu desenvolvimento, poderá desbloquear sua forma de pensar e assim ser mais criativa. As principais características de um *Design Thinker* segundo [Brown \(2008\)](#) são:

- Empatia: múltiplas perspectivas, ênfase nas pessoas, detalhistas, usam seus insights para inovar;
- Pensamento integrativo: criam soluções criativas a partir de ideias opostas;
- Otimismo: encaram os desafios na busca de soluções melhores;

- Experimentalismo: criam soluções inovadoras, não apenas ajustes adicionais na solução existente;
- Colaboração: trabalham em equipe, e além da interdisciplinaridade entre seus membros, muitos possuem experiência significativa em mais de uma área profissional.

2.2.2 Etapas do Design Thinking

Segundo [Vianna \(2012\)](#), as fases necessárias para aplicar o DT são quatro: *imersão*, fase onde deve-se mergulhar no problema; *análise*, conhecer o cenário; *ideação*: pensar fora da caixa para criar a solução; e *prototipação*, tirar a ideia do papel, ou seja, transformar o abstrato no físico, com o objetivo de representar a realidade, mesmo que de forma simplificada. Apesar de serem apresentadas de forma linear, as etapas do DT podem ser realizadas em qualquer sequência e configuradas conforme necessidades do projeto.

Imersão

Nesta primeira fase, o objetivo principal é entender bem o problema. Então deve-se imergir no universo do cliente ou negócio e visualizar o problema por diversas perspectivas ([VIANNA, 2012](#)). Esta etapa é muito importante para identificar realmente quais são as necessidades que o produto ou serviço devem oferecer. As atividades usadas na imersão podem ser divididas em dois grupos, a imersão preliminar, onde tem a pesquisa desk, pesquisa de soluções similares e pesquisa exploratória; e a imersão em profundidade, com entrevistas e cadernos de sensibilização.

Análise e Síntese

Todas as informações coletadas na imersão devem ser analisadas e organizadas nesta etapa. O objetivo é encontrar os padrões associados ao problema e definir as oportunidades que possam gerar soluções inovadoras na etapa seguinte ([VIANNA, 2012](#)). As técnicas usadas nesta etapa estão relacionadas com a criação de vários itens que irão ajudar na análise das informações. Os itens mais utilizados são: personas, mapa de empatia, mapa conceitual, cartões de insights, diagrama de afinidades e critérios norteadores.

Ideação

Tem como objetivo a criação de ideias inovadoras para resolver o problema, e para isso, várias ferramentas são utilizadas como mapas mentais, brainstorming, desenhos e outras atividades que estimulem a criatividade. Segundo [Vianna \(2012\)](#),

quanto mais ideias forem listadas e discutidas maior a probabilidade de chegar a uma solução inovadora e que atenda aos requisitos do problema.

Prototipação

As atividades usadas nesta etapa servem para tangibilizar as ideias geradas na ideação. [Vianna \(2012\)](#) afirma que prototipação pode ser definida como a transformação do abstrato no físico com o objetivo de representar a realidade. Porém sabe-se que a realidade pode ser bem complexa, então um protótipo pode ser elaborado de forma mais simples. Os principais resultados esperados nesta fase são as validações das ideias por meio da aplicação de testes de usabilidade do produto através do protótipo.

2.3 Extração, Transformação e Carga de dados - ETL

O termo ETL (*Extraction-Transformation-Loading*) pode ser definido, segundo [Mali e Bojewar \(2015\)](#), como um processo utilizado para extrair os dados de várias fontes, depois transformá-los usando ferramentas de limpeza e carregá-los em um banco de dados de destino.

Além de ser um dos processos mais importantes e complexos na construção de um Data Warehouse, o ETL também é bastante utilizado na migração de dados entre sistemas e integração de dados. No desenvolvimento de projetos com processos ETL, normalmente são utilizadas ferramentas capazes de extrair dados de diversas fontes, como arquivos de texto, planilhas, bancos de dados relacionais, além de outras. Esta extração geralmente é feita para uma camada intermediária onde ocorrem alguns ajustes através da limpeza e transformação, e na sequência, os dados são carregados na base de destino ([WU, 2010](#)).

Conforme dito em [Ross e Kimball \(2013\)](#), no momento de projetar a solução é necessário verificar vários itens, tais como: o tipo de dado da origem e suas restrições; as linguagens de programação que poderão ser utilizadas; as ferramentas ETL disponíveis para tal arquitetura; e por fim, é importante também verificar as habilidades da equipe que irá implementar. Estas atividades contribuem para que o projeto seja melhor estruturado, qualidade esta fundamental em sistemas ETL.

Segundo [Kimball e Caserta \(2004\)](#), ETL é um assunto simples e complicado ao mesmo tempo. Por exemplo, nas etapas intermediárias, o sistema ETL se divide em vários subcasos, que dependem de suas próprias fontes de dados diversas, de cada regra de negócio, softwares legados e aplicativos de relatório de destino incomuns. O desafio então é manter-se com o foco na missão básica do sistema ETL, que é extrair os dados das origens, transformá-los e carregá-los no destino.

2.3.1 Etapas do Processo ETL

Wu (2010) afirma que um sistema ETL bem projetado deve ser capaz de extrair os dados da fonte de origem, garantir padrões de qualidade e consistência dos dados, ajustar os dados independente da fonte de origem e carregar os dados ao destino pronto para serem consumidos pelos usuários finais.

Na Figura 2 pode ser verificada a estrutura geral para os processos ETL. Então, os dados podem ser extraídos de diferentes fontes de dados, em seguida vem a etapa da transformação, que pode utilizar uma base intermediária (*staging area*) para ajudar no ajuste e limpeza dos dados, e por fim, carregar estes dados no banco de dados de destino.

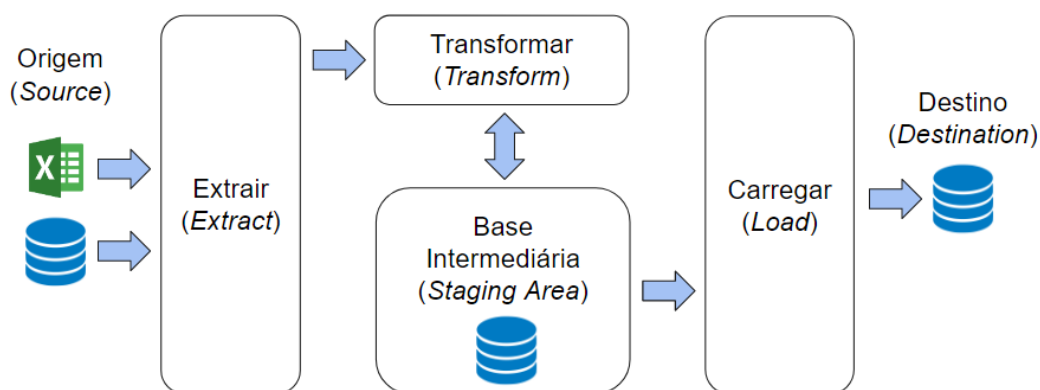


Figura 2 – Estrutura geral para processos ETL.
Fonte: (EL-SAPPAGH, 2017)

Extrair

Segundo El-Sappagh, Hendawi e Bastawissy (2017), a primeira etapa em qualquer processo ETL é a Extração dos dados das origens. Cada fonte de dados possui suas próprias características, então essa extração deve ser realizadas de forme que seja eficaz para o processo ETL.

Transformar

A segunda etapa, da transformação de dados, tem o objetivo de limpar e ajustar os dados de entrada para que estejam adequados conforme necessidade das etapas seguintes (MALI; BOJEWAR, 2015).

Base Intermediária (*Staging Area*)

Conforme características do projeto os dados extraídos a partir da origem podem ser tratados e limpos antes de serem guardados temporariamente em uma base

intermediária (*staging area*). Na sequência, estes dados gradados devem passar por mais algumas transformações antes de serem carregados na base de destino (ALI; MOHAMED, 2018).

Carregar (Load)

Segundo El-Sappagh, Hendawi e Bastawissy (2017), a etapa de carga consiste em popular a base de destino após as transformações realizadas nas etapas anteriores.

2.3.2 Ferramentas ETL

Atualmente há várias plataformas para se trabalhar com os processos ETL. Cada uma com suas vantagens e desvantagens. As mais conhecidas são: *Pentaho Data Integration*, *SQL Server Integration Services (SSIS)* da Microsoft, *Talend Open Studio* e a ferramenta de integração de dados da Oracle. Como a implementação do Flowup já é feita em um *software* da Microsoft, foi interessante utilizar o SSIS como ferramenta de ETL. Além disso, em várias pesquisas, aparece como uma ferramenta com muitos recursos de transformação de dados e integração com os vários tipos de arquivos que podem ser configurados para extrair os dados na primeira etapa do ETL.

Ferramentas ETL versus Codificação Manual (*Hand Coding*)

A escolha da ferramenta ETL a ser usada no projeto é uma decisão importante a ser tomada para a implementação do processo ETL. Geralmente, caso haja mudança na arquitetura escolhida, todo o projeto deverá ser implementado do zero. Por este motivo, a arquitetura deve ser construída de forma estruturada e consistente. Kimball e Caserta (2004) citaram duas possíveis formas de implementar as etapas do processo: utilizando uma ferramenta ETL, ou codificando manualmente (*hand coding*). Cada abordagem possui vantagens em relação à outra, cabe ao projetista decidir após uma análise detalhada.

Então, como apontam Kimball e Caserta (2004), se o uso de ferramentas ETL for a opção escolhida, o sistema será beneficiado com várias vantagens como: maior simplicidade na implementação, mais rapidez e economia também, pois o custo da ferramenta será compensado por um menor tempo de implantação; menor restrição quanto aos conhecimentos técnicos das pessoas que irão implementar o sistema; facilidade no tratamento dos dados, pois possuem uma grande variedade de conversores para a maioria dos tipos de dados existentes e que precisam ser transformados antes de serem enviados para a base de destino; em sua maioria, bom desempenho na sua execução mesmo se o conjunto de dados for grande; balanceamento da carga entre servidores, evitando a sobrecarga na transferência dos dados; dentre outras.

Por outro lado segundo [Kimball e Caserta \(2004\)](#), a abordagem Codificação Manual (*Hand Coding*) também possui suas vantagens, as principais são: disponibilidade de implementação de ferramentas de teste de unidade automatizadas, dessa forma contribuindo no controle de qualidade dos sistema; etapa de transformações com maior consistência para validações, relatórios de erros e atualização de metadados, tornando o sistema mais flexível; maior liberdade para gerenciar os metadados, porém todas as interfaces necessárias precisarão ser implementadas; liberdade também para utilizar a linguagem que a equipe tenha maior afinidade, e principalmente, se for o caso, continuar usando a mesma linguagem do sistema legado; e para fechar a lista das principais, a abordagem possui a flexibilidade para fazer o que for necessário no código, sem as limitações existentes em uma ferramenta já pronta.

2.3.3 SQL Server Integration Services - SSIS

O SQL Server Integration Services (SSIS) da Microsoft é uma plataforma que contém várias ferramentas de integração de dados incluindo funcionalidades prontas para serem utilizadas em tarefas comuns relacionadas a ETL. As principais tarefas e mais usadas durante um processo ETL, apresentadas em [Rodrigues, Coles e Dye \(2012\)](#), são:

- Extração - permite extrair dados de várias fontes de arquivos mais simples como texto, XML, planilhas do Microsoft Excel, mas também conseguem de bancos de dados relacionais e não relacionais;
- Validação - geralmente os dados extraídos precisam ser ajustados e/ou convertidos para que possuam o tipo exato do campo de destino. Então é possível padronizar ou limitar os dados no meio do processo;
- Limpeza - realiza alterações necessárias para adequação dos dados à base de destino, por exemplo, os campos relacionados a gênero podem ser “Masculino” ou “Feminino”, porém na base de destino, o campo mapeado exige que seja “1” ou “0”, respectivamente, e para isso é necessário implementar uma lógica que faça esse ajuste;
- Remoção de registros duplicados - esta tarefa permite que vários campos sejam verificados no mesmo registros, ou mesmo verificar apenas alguns campos específicos;
- Carregamento - possibilidade de carregar os dados de vários tipos de destino como arquivos ou bancos de dados.

Pacote da ferramenta de extração, transformação e carga - Pacote ETL

Através de um ambiente gráfico para implementação dos projetos, o SSIS disponibiliza várias opções para a criação e depuração dos pacotes. Com descrito em [Knight, Knight, Moss, Davis e Rock \(2014\)](#), a plataforma permite a criação de vários pacotes e cada pacote possui um fluxo de controle (*ControlFlow*), também conhecido como fluxo de trabalho ou fluxo de tarefa. Este por sua vez, possui um conjunto de tarefas (*ControlFlowTasks*) e contêineres, Figura 3.

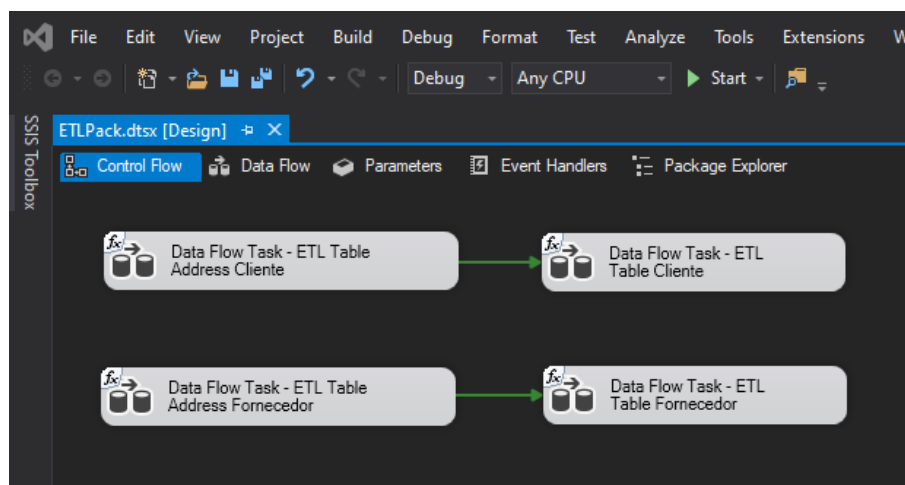


Figura 3 – Exemplo de um Fluxo de Controle e alguns fluxos de dados adicionados
Fonte: Elaborado pelo autor.

ControlFlow Tasks utilizadas neste trabalho

Nas subseções deste item são descritas as principais funcionalidades das tarefas utilizadas na ferramenta proposta.

DataFlow Task

Dentre as tarefas do *ControlFlow* esta é especial. Ela possui um fluxo de dados (*DataFlow*) para implementação dos processos ETL. Na Figura 3, pode-se verificar o ambiente do fluxo de controle com quatro fluxos de dados adicionados, e as conexões existentes entres eles.

Como apresentado em [Knight, Knight, Moss, Davis e Rock \(2014\)](#), o fluxo de dados (*DataFlow*) possui vários componentes para tratamento e manipulação dos dados, que podem ser de três tipos. O primeiro tipo, Fontes (*Sources*), corresponde aos componentes que têm a função de se comunicar com as fontes de origem e extrair seus dados para dentro do fluxo de dados. Em seguida tem os tipos Transformações (*Transformations*) e Destinos (*Destinations*).

Estes tipos de componentes disponíveis no fluxo de dados representam fielmente as etapas do processo ETL, e isso facilita sua utilização por usuários iniciantes na ferramenta. Podem ser vistos na Figura 4 alguns componentes no ambiente do fluxo de dados. E na Figura 5, tem um exemplo de tradução de valores no campo “TipoFornecedor”, então quando o valor extraído da fonte for “Jurídico” o valor encaminhado ao destino é “1” conforme operação ternaria implementada.

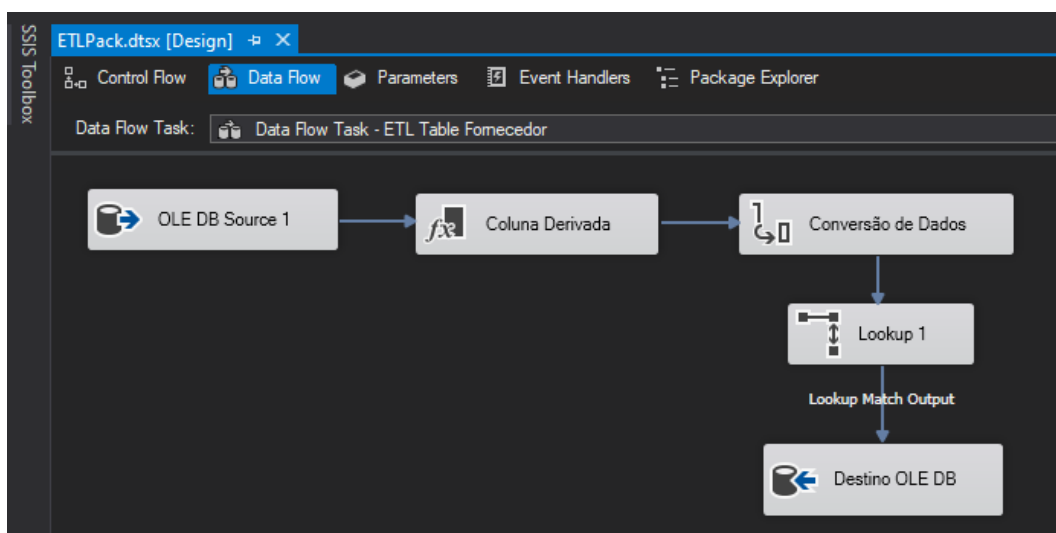


Figura 4 – Exemplo do ambiente de implementação/configuração dos componentes de um Fluxo de Dados. Fonte: Elaborado pelo autor.

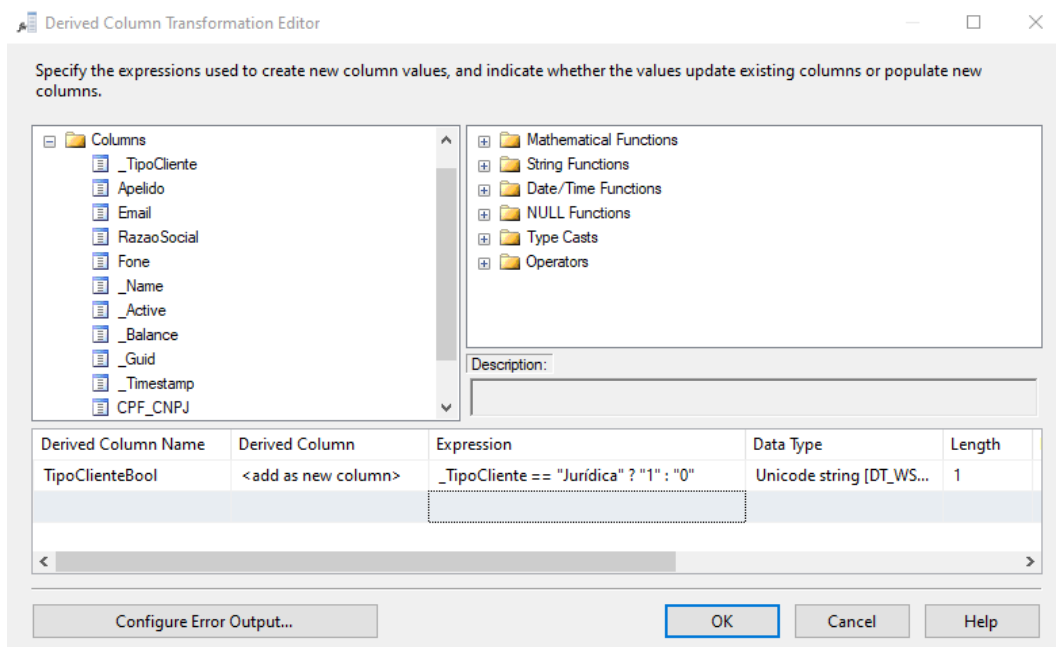


Figura 5 – Exemplo de configuração do componente *Derived Column*. Fonte: Elaborado pelo autor.

Execute SQL Task

Esta tarefa pode ser usada em várias situações, por exemplo, truncar uma tabela importação, consultar a contagem das linhas de um processo e usar esses dados para uma necessidade do projeto. É possível ainda executar instruções SQL parametrizadas, como apresentado na Figura 6.

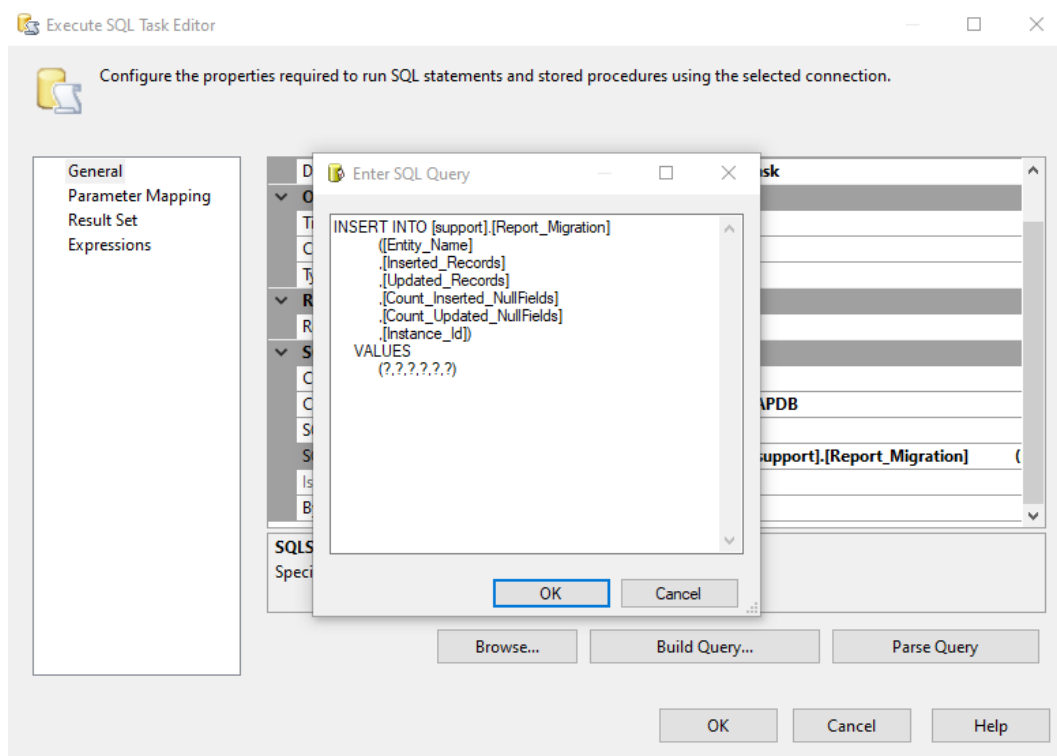


Figura 6 – Configuração da tarefa *Execute SQL*
Fonte: Elaborado pelo autor.

Principais Componentes de um DataFlow (Fluxo de Dados)

O *DataFlow* é composto por três tipos de componentes: origens, transformações e destinos. Que são exatamente os fundamentos de um processo ETL. As origens extraem os dados de arquivos simples como bancos de dados *OLE DB* e outros locais; as transformações ajustam os dados; e os destinos realizam as inserções dos dados na base de destino. Todos estes componentes estão explicados com detalhes em [Knight, Knight, Moss, Davis e Rock \(2014\)](#) e os principais são descritos nas subseções abaixo.

Esse processamento de dados realizado na execução do pacote ETL é feito na memória do servidor SSIS. Isso explica a velocidade do processo ETL usando esta ferramenta, pois é bem mais rápido realizar as transferências de dados diretamente na memória do que usar tabelas temporárias, chamadas de tabelas de preparação

(*Staging tables*). Por isso, a quantidade de memória do servidor SSIS deve ser escolhida de forma que suporte a quantidade de dados máxima estimada para o projeto (KNIGHT, 2014).

Origens (Sources)

Também chamados de componentes fontes, são os componentes responsáveis por especificar a localização do dado de origem do processo. Os principais componentes deste tipo disponíveis no SSIS, como descreve Knight, Knight, Moss, Davis e Rock (2014) são:

- *OLE DB Source* - É o componente de origem mais comum; ele pode ser configurado para qualquer origem de dados compatível com OLE DB, como por exemplo, SQL Server, Oracle ou *IBM DB2*, banco de dados desenvolvido pela IBM;
- *Excel Source* - Este componente pode apontar para arquivos do Excel e assim utilizar os dados da planilha no processo. Uma característica importante é que o SSIS converte os campos ou colunas do Excel que estão no formato geral em uma *string unicode*, ou *nvarchar* no SQL Server. Então se não é o tipo de dado que o campo de destino está esperando, será necessário usar algum componente de transformação;
- *Flat File Source* - Este componente tem como entrada arquivos de texto ou dados delimitados, geralmente delimitados por vírgula (CSV). Em sua configuração deve-se escolher como a tabela será montada de acordo com as colunas existentes no arquivo.

Transformações Comuns

Estes componentes geralmente são conectados à saída de um componente de origem, de forma direta ou indireta. Este último corresponde ao caso onde há um ou mais componentes entre o componente de origem e o componente em questão. Os principais componentes de Transformação no SSIS, segundo Knight, Knight, Moss, Davis e Rock (2014) são:

- *Derived Column* - Neste componente uma nova coluna é adicionada ao processo. O valor dessa coluna pode ser calculada por uma lógica implementada em um *script*, onde os valores de outras colunas podem ser adicionados a esta lógica, inclusive valores dos parâmetros adicionados no pacote;

- *Data Conversion* - Na configuração deste componente aparecem todas as colunas em uma tabela. Cada coluna é configurada individualmente, então se houver necessidade a coluna é adicionada à lista de conversões onde deve-se escolher o novo tipo de dado em uma lista de seleção, e ainda deve-se definir um apelido (*alias*) para a nova coluna;
- *Conditional Split* - Este componente permite criar vários caminhos para o processo conforme lógicas específicas configuradas no componente;
- *Lookup* - Permite que uma pesquisa seja realizada em uma tabela, retornando uma coluna. Neste caso tem-se a primeira tabela, que está conectada a um componente de origem, e tem agora a segunda tabela, que é adicionada na configuração deste componente. Então as colunas que serão pesquisadas devem ser selecionadas na primeira tabela, e associadas às colunas correspondentes na segunda tabela, onde as colunas que devem retornar devem ser selecionadas, como pode ser visto na Figura 7. Na saída do componente são gerados dois caminhos possíveis, *Match*, caso as colunas pesquisadas sejam encontradas, e *No Match*, caso em que não são encontradas. Este componente ainda pode ser configurado de forma que não retorne nenhuma coluna, e nestes casos o objetivo é apenas a pesquisa, e o retorno é apenas a indicação de que a pesquisa deu *Match* ou não. Por fim, o componente *Lookup* pode ainda ser configurado de forma a ignorar o resultado da pesquisa, então nestes casos mesmo que as colunas pesquisadas não sejam encontradas, o fluxo do processo irá continuar pelo caminho do *Match*;
- *Row Count* - Permite a contagem dos registros que passam por este componente. O dado da contagem é armazenado em uma variável do pacote;
- *Script Component* - Permite a criação de *scripts* .NET (VB ou C#) personalizados. Eles podem ser usados como transformações, origens ou destinos. São muito usados para validar os dados de entrada como CPF, CNPJ, CEP, e outros mais. É possível ainda criar um componente personalizado, adicionando as entradas e calculando as saídas.
- *Union All* - Tem como entrada a saída de vários componentes, e sua saída é a combinação de todas estas entradas, resultando em um único conjunto contendo todas as colunas reunidas.

Outras Transformações

Dentre os vários componentes nesta categoria de Outras Transformações citados em Knight, Knight, Moss, Davis e Rock (2014), seguem os componentes usados neste trabalho:

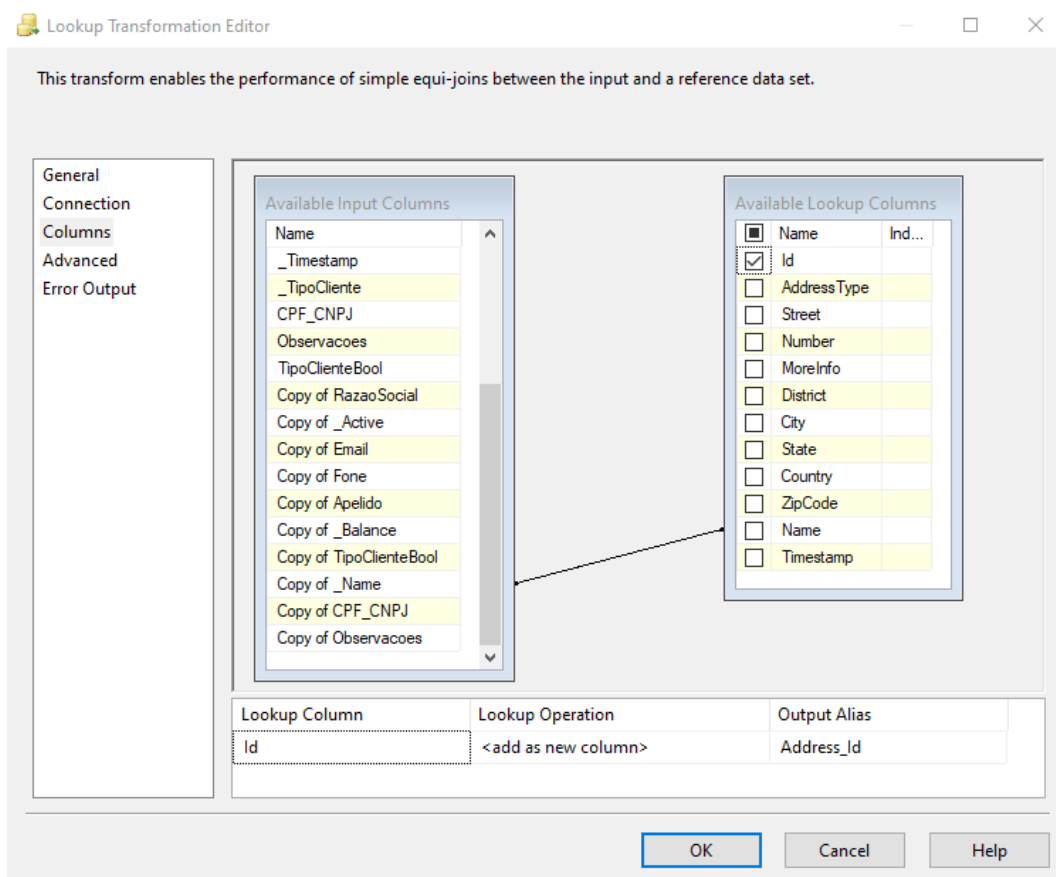


Figura 7 – Exemplo de configuração do Componente *Lookup*

Fonte: Elaborado pelo autor.

- *Multicast* - Permite a distribuição de uma única entrada para vários caminhos;
- *OLE DB Command* - Tem a função de executar uma instrução SQL para cada registro de entrada, ou seja para cada linha do fluxo do processo. Na configuração deste componente, podem ser definidos vários parâmetros para compor a instrução SQL. Este componente é muito usado para realizar atualizaçõesmos dados das tabelas de destino.

Destinos (Destinations)

Nos componentes de destino, as colunas recebidas como entrada pelo fluxo devem ser associadas às respectivas colunas na base de destino configurada no componente. Estas associações devem ser realizadas na opção Mapeamentos (*Mappings*). Vários componentes de destino foram descritos em [Knight, Knight, Moss, Davis e Rock \(2014\)](#), seguem os mais usados:

- *OLE DB Destination* - É o componente de destino mais comum; ele pode ser configurado, assim como no caso da origem, para qualquer origem de dados compatível com OLE DB, como Oracle, IBM DB2, SQL Server, dentre outros.

- *Excel Destination* - Assim como o *Excel Source*, aponta para arquivos do *Excel*, só que no caso deste componente, em vez de ter como entrada uma planilha, ele recebe os dados de um componente qualquer do fluxo e envia em sua saída os dados para alimentar uma planilha, a mesma que está configurada nele;
- *Flat File Destination* - Envia os dados recebido pelo fluxo para um arquivo simples, que pode ser de largura fixa ou delimitado, por vírgula geralmente (CSV).

2.3.4 Objetos Dinâmicos no Pacote SSIS

Para projetos mais simples, onde não é necessário alterar nenhum valor, os componentes do SSIS são configurados com valores fixos, como por exemplo, o nome de uma tabela a ser carregada, um valor que se queira pesquisar em uma tabela, entre outros. Por isso, conforme apresentado em [Knight, Knight, Moss, Davis e Rock \(2014\)](#), há alguns objetos no SSIS que podem ser usados para criar pacotes dinâmicos, ou seja, estes objetos podem ser alterados com o pacote em execução. São três objetos que podem trabalhar em conjunto para criar um pacote dinâmico: Expressões, Variáveis e Parâmetros.

Expressões

Basicamente, pode-se criar uma expressão em alguns componentes de um *DataFlow Task* utilizando os valores de variáveis, parâmetros e vários tipos de funções (Figura 8) que são disponibilizadas no ambiente de criação da expressão.

Variáveis

Segundo [Knight, Knight, Moss, Davis e Rock \(2014\)](#), este objeto pode ser criado livremente para uso do pacote, podendo ser apontado em algum componente. E seu valor pode ser uma composição de outras variáveis e parâmetros com a opção também de usar vários tipos de funções. O mais importante é que o seu valor pode ser alterado durante a execução do pacote. Na Figura 8 são apresentados alguns exemplos de variáveis criadas, e na janela aberta (*popup*) pode-se ver a composição de uma expressão para calcular o valor de uma variável específica a partir do valor de outras variáveis.

Parâmetros

Os parâmetros no SSIS têm um uso similar ao das Variáveis, porém estes objetos possuem algumas diferenças muito importantes para a implementação do processo ETL. Uma é que o valor de um parâmetro não pode ser alterado durante a execução do pacote. A outra diferença é que quando o *Deploy* do pacote (explicado na Seção 2.3.4)

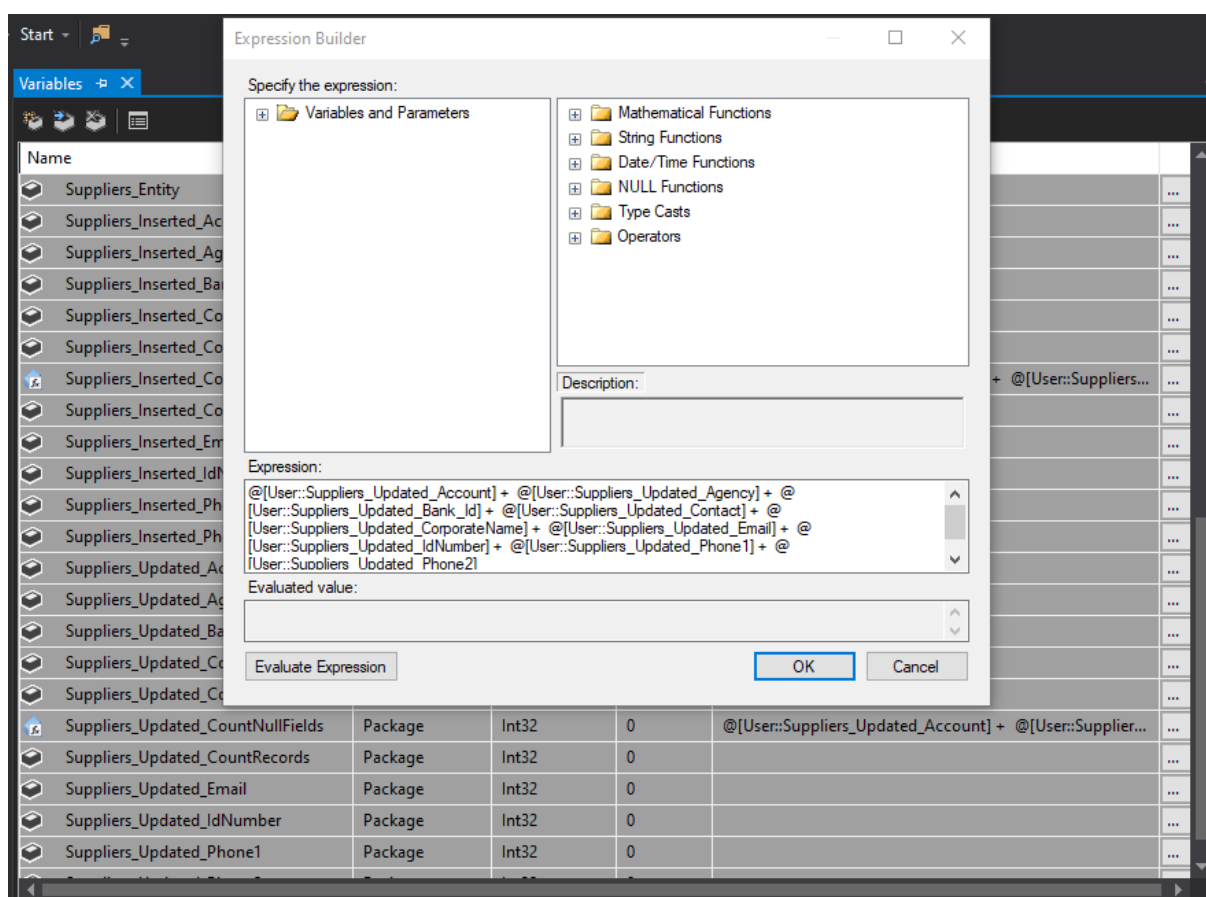


Figura 8 – Criação de variáveis no pacote SSIS

Fonte: Elaborado pelo autor.

é realizado para o Catálogo SSIS (mais detalhes na Seção 2.3.4) do SQL Server, os parâmetros são automaticamente disponibilizados na configuração do SSISDB, porém as variáveis não, estas são de uso interno do pacote.

Como apresentado em Knight, Knight, Moss, Davis e Rock (2014), os parâmetros podem ser de nível de projeto ou de pacote. Os parâmetros do projeto são mais externos e podem atribuir valores em mais de um pacote. Os parâmetros do pacote são restritos ao pacote em que foram criados (Figura 9) e podem alterar os valores apenas de seus componentes internos.

Na configuração do *Environment* do Catálogo SSIS podem ser criadas algumas variáveis e mapeá-las com os parâmetros do pacote, dessa forma, uma aplicação externa pode enviar outros valores para os parâmetros e estes serão aplicados na execução do pacote ETL.

Catálogo SSIS

Ao final da implementação do pacote ETL no SSIS, ele precisa ser implantado no SQL Server, mais especificamente no Catálogo SSIS. Segundo Tok, Parida, Masson e Ding (2012), nas versões anteriores do SSIS os pacotes eram executados no

Name	Data type	Value	Sensitive	Required	Descri
TableAddress	String	[empresa1].[Address]	False	False	
TableClients	String	[empresab].[Clients]	False	False	

Figura 9 – Criação de parâmetros no pacote SSIS
 Fonte: Elaborado pelo autor.

ambiente do *desktop* através de ferramentas de linha de comando, como *Dtexec.exe*. A partir da versão de 2012 a *Microsoft* criou um novo aplicativo para o *SQL Server* chamado de Catálogo SSIS.

O Catálogo SSIS é um servidor totalmente funcional. Ele possui um banco de dados dedicado chamado de SSISDB, que tem muitas visualizações (*Views*) públicas. Assim, as informações da implantação ou execução podem ser consultadas por qualquer ferramenta que use *scripts* SQL (DAS, 2020).

Das (2020) afirma que é importante entender a arquitetura do banco de dados e os vários objetos dentro do Catálogo. O diagrama mostrado na Figura 10 explica um pouco a arquitetura do banco de dados do catálogo e como ele se relaciona com o mecanismo de banco de dados do *SQL Server*.

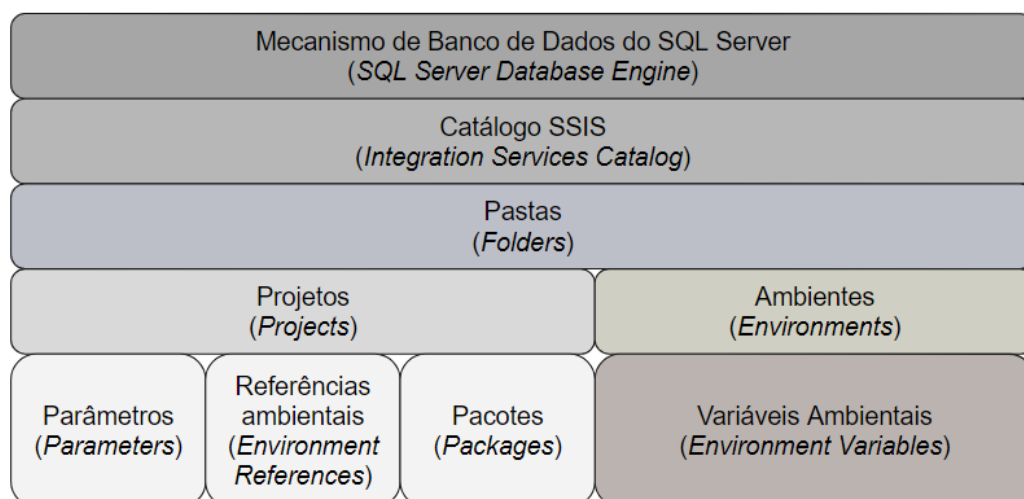


Figura 10 – Arquitetura da base de dados do Catálogo SSIS
 Fonte: (DAS, 2020)

O Catálogo SSIS permite que usuários com perfil de administrador desenvolvam, implantem e gerenciem aplicativos de integração de dados. Ele possui um conjunto de ferramentas que auxilia no desenvolvimento de projetos e pacotes do SSIS. Entre estas ferramentas estão os procedimentos armazenados (*stored procedures*), as visualizações (*views*) e as funções com valor de tabela (*table-valued functions*) (TOK, 2012).

Implantação do pacote no *SQL Server* (*Deploy*)

Antes de realizar o *Deploy* pelo SSIS, deve-se criar o Catálogo SSIS dentro do Catálogo SSIS (*Integration Services Catalogs*) no ambiente de configuração do SSMS. Durante este processo, a base SSISDB é criada automaticamente, deixando o ambiente pronto para receber a implantação.

Segundo Verbeeck (2015), a partir da versão 2012 do SSIS, foi adicionado o passo de implantação de projeto, assim o projeto inteiro poderia ser implantado no Catálogo SSIS configurado no *SQL Server*. Na versão de 2016 foi introduzido o passo de implantação de pacote, permitindo que um ou mais pacotes fossem implantados de forma incremental (VERBEECK, 2015).

Execução do pacote ETL

Na fase de testes do processo ETL, o pacote pode ser executado no próprio ambiente de desenvolvimento do Visual Studio, dessa forma é possível verificar o fluxo de execução passando por todos as tarefas e componentes implementados, como mostrado na Figura 11.

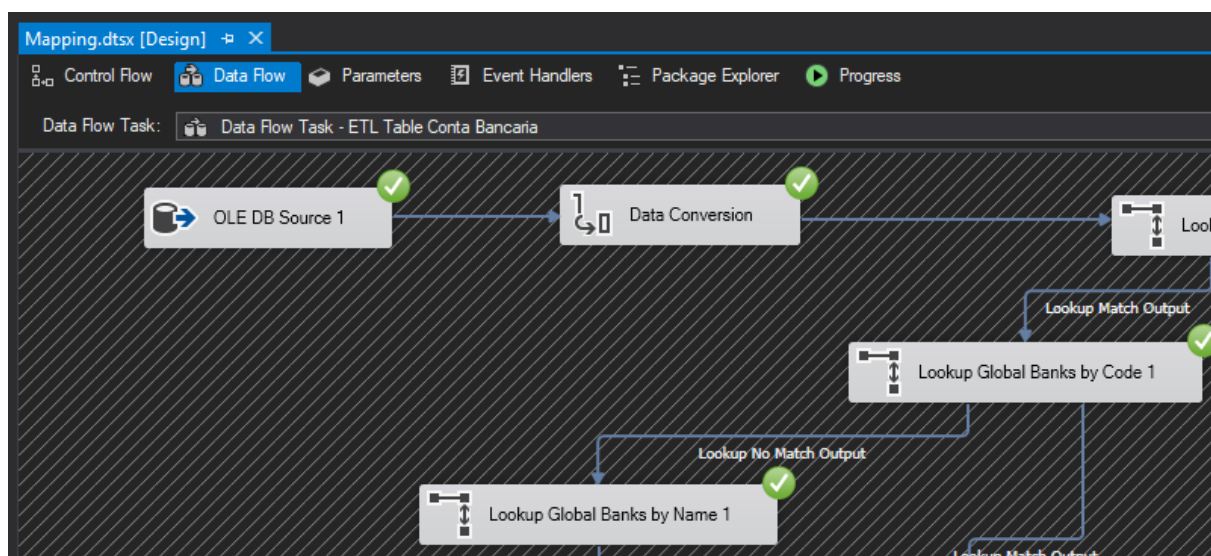


Figura 11 – *DataFlow* na Execução do pacote ETL no modo de depuração (*debugging*)
Fonte: Elaborado pelo autor.

Ao executar um pacote pelo ambiente de gerenciamento do Catálogo SSIS no SSMS é possível visualizar relatórios bem detalhados com todos os diagnósticos referentes ao processo executado, dados sobre a performance e uma lista de várias mensagens informando detalhes sobre cada componente.

Porém, para executar um pacote no SSIS através de uma aplicação web, por exemplo, deve ser usado o *T-SQL*, que é uma extensão do SQL. O código padrão para

executar um pacote usando T-SQL é mostrado na Figura 12, que pode ser gerado na configuração do pacote ETL (KNIGHT, 2014).

```
SQLQuery2.sql - (...-37108VH\FAST (52))* -# X
[Declare @execution_id bigint
EXEC [SSISDB].[catalog].[create_execution] @package_name=N'Mapping.dtsx', @execution_id=@execution_id OUTPUT,
                                           @folder_name=N'MigrUpPacks', @project_name=N'MigrUp ETL',
                                           @use32bitruntime=False, @reference_id=2, @runinscaleout=False
Select @execution_id
DECLARE @var0 smallint = 1
EXEC [SSISDB].[catalog].[set_execution_parameter_value] @execution_id, @object_type=50,
                                                       @parameter_name=N'LOGGING_LEVEL',
                                                       @parameter_value=@var0
EXEC [SSISDB].[catalog].[start_execution] @execution_id
GO
```

Figura 12 – Código de execução padrão gerado na configuração do pacote ETL
Fonte: Elaborado pelo autor.

Além disso, os sistemas podem precisar alterar alguma configuração em algum componente no modo de execução. Para isso, segundo Microsoft (2017), pode ser adicionado um código com a estrutura mostrada na Figura 13. Então, os parâmetros adicionados no pacote ETL poderão ser associados às variáveis criadas no *Environment* do Catálogo SSIS, na pasta SSISDB.

```
SQL Copiar
catalog.set_environment_variable_value [ @folder_name = ] folder_name
, [ @environment_name = ] environment_name
, [ @variable_name = ] variable_name
, [ @value = ] value
```

Figura 13 – Código adicional para execução do pacote ETL com parâmetros
Fonte: (MICROSOFT, 2017)

2.4 Solução alternativa

Nesta seção foram pesquisadas algumas soluções alternativas ao processo ETL para a resolução do problema. Porém, foi encontrada apenas uma, que está descrita na subseção abaixo.

2.4.1 *Extract Load Transform* (ELT)

Em comparação com o processo *Extract Transform Load* (ETL), Hussain (2019) afirma que o processo *Extract Load Transform* (ELT) ganha no quesito velocidade nas duas primeiras etapas, pois extrai os dados da origem e carrega tudo em um repositório de dados armazenados (*data lake*). Dependendo do projeto, a etapa do carregamento pode também realizar pequenas transformações simples. E então, na etapa da transformação, conforme a demanda dos usuários, os dados são transformados para

relatórios analíticos. Este processo já era usado na prática em algumas empresas, mas com o aumento do poder de processamento cada vez maior da computação em nuvem em conjunto com o surgimento do *Hadoop*, o ELT está se tornando cada vez mais conhecido. Na Figura 14 é apresentado um processo ELT típico.

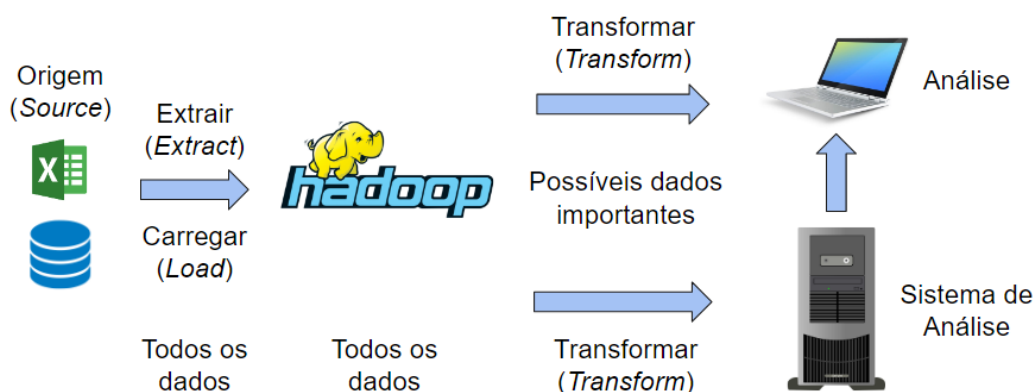


Figura 14 – Processo ELT típico

Fonte: (JAVAPOINT, 2021)

Com a crescente diversificação dos dados de origem, os processos ELT tem a vantagem de não se limitar a extração de dados relacionais, pois o *Hadoop* pode processar dados semiestruturados e não estruturados (HUSSAIN, 2019).

Porém, o processo ELT não é viável para o desenvolvimento deste trabalho, pois, logo após a extração dos dados da origem, existe a etapa de mapeamento entre as bases, que deve ser realizada pelo próprio usuário. E neste caso, a carga na base de destino deve ser realizada apenas após este mapeamento, isto é, a carga não pode ser feita junto com a extração, como é o caso do processo ETL. Por isso, o processo usado neste trabalho deve ser o ETL.

3 Método

Este capítulo descreve as etapas do processo de desenvolvimentos do Migrup em suas respectivas seções: etapa do *Design Thinking* na Seção 3.1; etapa de desenvolvimento na Seção 3.2; e, as ferramentas utilizadas na Seção 3.3. Cada uma delas descrevendo todos os passos utilizados na implementação da etapa correspondente.

O processo de desenvolvimento deste trabalho foi baseado na metodologia do *Design Thinking* (DT). Isso foi feito desde o entendimento do problema e definição dos objetivos, até as validações conseguidas durante os processos de prototipação e implementação do projeto. Esta escolha foi feita na tentativa de construir uma solução criativa e que atendesse às necessidades do usuário.

Atualmente, conforme mostrado na Figura 15, novos clientes do Flowup precisam preencher, manualmente, uma planilha pré-formatada pela equipe técnica. Essa planilha contém as abas específicas e os campos que podem ser cadastrados no Flowup. A ferramenta atual importa esses dados dessa planilha e os insere na base de dados do Sistema.

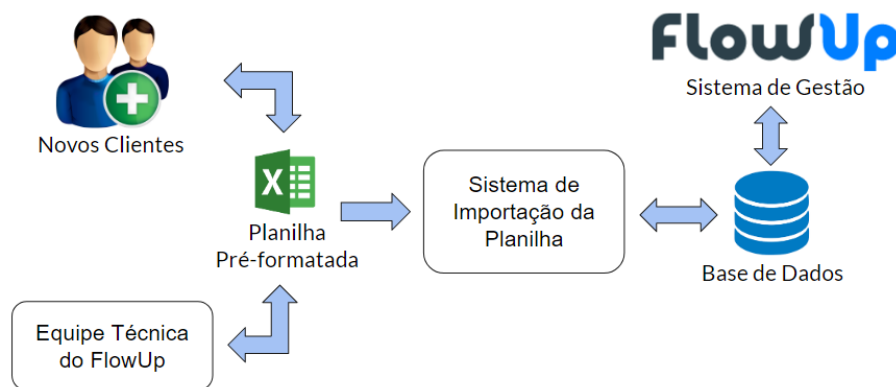


Figura 15 – Fluxo da ferramenta de importação atual do Flowup
Fonte: Elaborado pelo autor.

Sendo assim, a ferramenta proposta neste trabalho, chamada de Migrup, foi desenvolvida para suprir algumas necessidades apresentadas pela ferramenta legada e facilitar o processo de importação, adicionando novas funcionalidades.

As entidades do sistema Flowup já importadas pela ferramenta legada são: Clientes, Fornecedores, Contas, Receitas/Despesas e Transferências. Então, a princípio, estas mesmas entidades também foram adicionadas ao Migrup.

Este trabalho foi organizado em etapas e subetapas, apresentadas na figura 16. As etapas gerais foram divididas e sequenciadas conforme seus processos e objetivos no projeto. Então, primeiro vem a etapa do *Design Thinking*, que resulta na geração

de informações fundamentais através das ferramentas em suas subetapas. Estas informações foram usadas no momento seguinte, etapa do Desenvolvimento, para o início da implementação da ferramenta proposta neste trabalho. Nesta, todos os passos necessários para realizar a implementação foram definidos. Após isso, na terceira etapa, a ferramenta desenvolvida foi implantada a princípio em um servidor local.

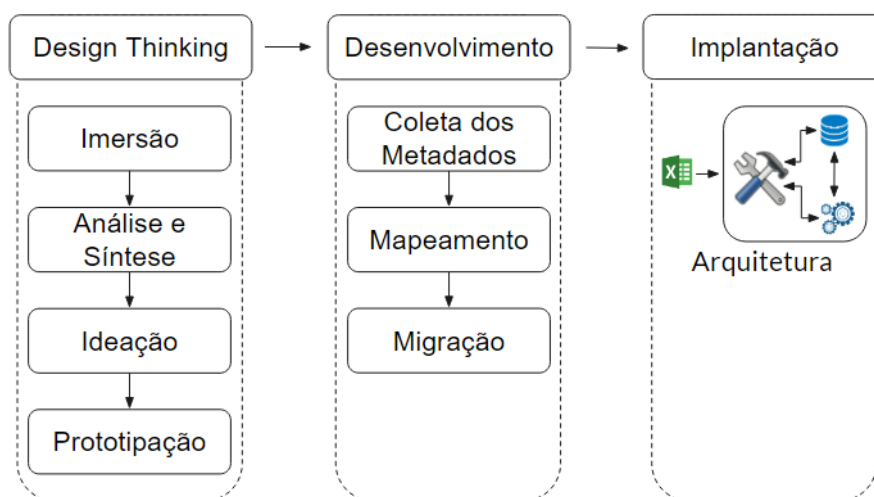


Figura 16 – Fluxo das etapas utilizadas no desenvolvimento do Migrup

Fonte: Elaborado pelo autor.

3.1 *Design Thinking* - DT

Esta seção descreve como as ferramentas de cada etapa do DT foram aplicadas no desenvolvimento deste trabalho. Mas, os resultados de cada etapa do DT são apresentados no Capítulo 4.

3.1.1 Imersão

O problema a ser trabalhado foi definido após reunião com o *Scrum Master* do sistema Flowup junto ao *Product Owner*, onde foram passadas as informações e necessidades da ferramenta a ser desenvolvida. O problema já tinha sido identificado pelo *Scrum Master*, que percebeu a dificuldade no processo atual de importação dos dados de novos clientes. Este processo é realizado pelas equipes de suporte e marketing do Flowup, que utilizam a ferramenta legada de importação, descrita na Seção 2.1.

Nesta etapa do DT foram realizadas entrevistas individuais com os membros da equipe do Flowup: *Scrum Master*, Desenvolvedor de software Sênior, uma pessoa da equipe do suporte e uma do marketing, sendo um total de quatro entrevistas nesta fase. As informações coletadas nestas entrevistas foram documentadas em anotações,

áudios e respostas dos questionários. Estas informações foram fundamentais para o desenvolvimento deste trabalho, principalmente na fase inicial. As perguntas feitas à equipe do Flowup foram:

1. Qual sua opinião a respeito da ferramenta atual de importação dos dados de clientes?
2. Quais as principais dificuldades na importação dos dados do cliente, desde o início do processo até sua finalização?
3. Qual sua frequência de uso da ferramenta de importação atual? (Ex: 2 vezes por semana)

Entidades importadas atualmente: Clientes, Fornecedores, Contas, Receitas /Despesas, e Transferências.

4. Quais entidades importadas atualmente do Flowup são as mais importantes?
5. O que acha que falta melhorar na ferramenta atual?
6. Quais outras entidades do Flowup que poderiam ou deveriam ser importadas em uma nova versão da ferramenta de importação?
7. Esta nova ferramenta poderia ser utilizada pelos próprios clientes do Flowup?

Em seguida, foram pesquisadas algumas soluções similares existentes no mercado. Assim, foram encontrados alguns sistemas *Cloud ERP* que disponibilizam a seus clientes a funcionalidade de importar os dados de sua empresa. Para conferir o funcionamento das ferramentas, foi necessário criar uma conta temporária nos sistemas encontrados. Estas experiências foram anotadas de forma detalhada para apresentá-las nas etapas seguintes do DT.

Após a definição do problema e levantamento das funcionalidades necessárias à nova solução, foram realizadas várias pesquisas relacionadas à migração de dados, metodologias utilizadas e melhores abordagens a serem executadas em sistemas *Cloud ERP*.

O resultado destas pesquisas trouxe algumas questões técnicas que precisaram ser discutidas. Então, nesta etapa do DT, foram realizadas mais três reuniões. A primeira foi realizada com o orientador deste trabalho. E a segunda, foi realizada com o orientador e uma professora do DEINFO - UFRPE. Todas as informações coletadas nestas duas primeiras reuniões foram anotadas e agrupadas por assunto. Por fim, na terceira reunião, todas as sugestões anotadas foram discutidas com um Desenvolvedor Sênior do Flowup, que também contribuiu com algumas sugestões e possíveis so-

luções. Então, ainda nesta reunião, as principais ferramentas usadas no Migrup foram definidas após validação do Desenvolvedor Sênior.

Estas questões estavam relacionadas à linguagem de programação a ser usada, às abordagens de desenvolvimento e aos métodos utilizados para otimizar o processo de migração dos dados.

3.1.2 Análise e Síntese

Na etapa de análise e síntese o objetivo foi de definir oportunidades que pudessem agregar ideias à solução. Nesta etapa foram realizadas quatro reuniões ao todo. As três primeiras com o objetivo de apresentar e discutir as informações coletadas na etapa da imersão, e a última para validar as definições geradas após as discussões e levantamento das sugestões. A primeira foi realizada com o orientador deste trabalho; na segunda reunião, além do orientador teve a presença de um professor do DEINFO - UFRPE; e a terceira reunião foi realizada com um dos fundadores da Fast-Soluções, empresa desenvolvedora do Flowup. Todas as observações e sugestões coletadas nestas reuniões foram anotadas e organizadas. Estas informações geraram algumas definições para o desenvolvimento da ferramenta proposta, dessa forma, na quarta reunião, agora com o *Scrum Master* do Flowup, estas definições geradas foram discutidas de acordo com a viabilidade de implementação neste trabalho e em seguida validadas.

3.1.3 Ideação

O objetivo desta etapa é compartilhar os resultados obtidos nas etapas anteriores do DT, para tentar gerar ideias inovadoras à ferramenta proposta. Então, foi realizada uma reunião com a equipe do Flowup, e aplicada a técnica de *brainstorm*, que é uma abordagem de reuniões em grupo para desenvolver ideias criativa e originar soluções durante esta etapa. Nessa reunião com a equipe, as informações geradas nas etapas anteriores do DT foram utilizadas para apresentar melhor o problema aos participantes. Foras usadas as informações dos critérios norteadores gerados na etapa da análise 4.2 e as principais características dos sistemas similares descritos nos resultados da etapa da imersão 4.1.

3.1.4 Prototipação

Nesta última etapa do DT foi construído um protótipo com o objetivo de validar as ideias criadas na etapa anterior. Para elaboração do protótipo foi usado o software *Adobe XD* que permite o desenvolvimento de protótipos interativos.

O protótipo construído foi apresentado à equipe do Flowup, um membro por vez. Nestas reuniões, cada um foi orientado a tentar realizar o processo de importação dos dados navegando pelas telas; e conforme o processo caminhava, as observações e sugestões eram feitas pelo membro da equipe. Além disso, foram observadas as dificuldades encontradas durante este processo de manuseio do protótipo.

Durante cada reunião realizada nesta etapa, essas novas informações foram anotadas e organizadas, de modo que novas ideias surgiram para melhorar a interação do usuário com a ferramenta.

3.2 Desenvolvimento

O início da implementação ocorreu após a validação do protótipo com a equipe do Flowup. As primeiras ações realizadas foram pesquisar as ferramentas e abordagens que poderiam ser utilizadas no desenvolvimento deste trabalho. No projeto do Migrup, há três objetivos que podem ser considerados os principais do processo, que é a coleta dos metadados da origem e destino; o mapeamento dos campos entre as bases e as validações dos dados importados da planilha; e o terceiro objetivo, a migração dos dados para o destino.

Desse modo, o processo de desenvolvimento foi dividido em três etapas, cada uma correspondendo a um desses objetivos. A primeira etapa foi a coleta dos metadados da planilha (origem) e da base do sistema Flowup (destino). Estes dados são usados no painel de mapeamento, que corresponde à segunda etapa, e consiste na associação manual de cada campo das tabelas de destino com algum campo da planilha importada. Por fim, na terceira etapa, ocorre a migração dos dados para a base do sistema Flowup, conforme mapeamento dos dados.

As primeiras etapas, coleta dos metadados e mapeamento, foram implementadas no modo de codificação manual (*hand coding*), pois não se sabe como será o padrão da planilha importada, nem a quantidade de abas que possui. Dessa forma não seria viável usar a ferramenta ETL para importar uma planilha do Excel sem estar em um padrão específico. Essa foi uma melhoria da solução proposta, possibilitando a importação de qualquer tabela do Excel. Já a terceira etapa, da migração, foi utilizada uma ferramenta ETL para facilitar na implementação, principalmente no tratamento dos dados, como também na manutenção da ferramenta, além de melhorar o desempenho na maioria dos projetos e não gerar nenhuma despesa a mais, pois pode ser implantada no mesmo servidor *SQL Server* do sistema Flowup.

Esta seção descreve os métodos usados para implementar cada etapa da ferramenta proposta.

3.2.1 Coleta dos Metadados

Na planilha padrão de importação do Flowup há cinco abas, onde cada aba corresponde a uma entidade, porém, em cada aba há uma tabela. Foi interessante distinguir os termos *aba* e *tabela*, pois as abas podem conter alguma informação em seu cabeçalho antes de iniciar a tabela com os dados que serão importados. Isto vale também para as planilhas exportadas de outros sistemas de gestão, que poderão ser importadas pelo Migrup.

Esta subseção descreve os métodos usados para implementação da coleta dos metadados necessários.

Planilha Eletrônica

Os metadados da planilha correspondem aos nomes (títulos) das colunas, então, para coletar estes metadados da origem, foi necessário filtrar as linhas que contém os nomes das colunas de cada tabela (ou aba).

O Migrup permite a importação de qualquer planilha Excel, independente do número de abas. Mas para que a coleta dos metadados seja realizada corretamente, todas as abas devem ser padronizadas quanto à posição das linhas que contém os nomes das colunas das tabelas.

Base Intermediária

A base intermediária, MAPDB, representa a base de destino para a etapa do mapeamento. Esta base deve ser praticamente um espelho da base do Flowup (Cash), pois basta ter os campos que serão importados pela ferramenta. Então, esta coleta dos metadados da base MAPDB foi automatizada, principalmente para evitar alteração no código todas as vezes que a base Cash precisar ser alterada, e assim, toda vez que a migração é executada, estes metadados são consultados. As consultas à base MAPDB são realizadas para todas as tabelas do esquema *dbo*, e trazem os metadados solicitados.

3.2.2 Mapeamento entre as Tabelas

O processo completo do mapeamento foi dividido em três partes: a associação entres os campos da base de origem e destino; validação dos dados; e, inserção destes na base MAPDB. Esta subseção descreve os passos utilizados para poder implementar este processo.

Associação entres os Campos de Origem e Destino

O processo de mapeamento consiste em associar um campo da base de destino com uma coluna da planilha. Isso foi feito para possibilitar que qualquer planilha fosse importada, independente da ordem de suas colunas, como também dos nomes dessas colunas. Para isso, os metadados coletados na etapa anterior foram trabalhados e apresentados na interface gráfica de forma que, para cada campo da base Cash, qualquer coluna da planilha pudesse ser escolhida pelo usuário.

Para implementar isso na interface gráfica, foi preciso agrupar os metadados coletados da base MAPDB por tabela, estes grupos foram usados para preencher os dados das tabelas da base de destino. Já os metadados coletados da planilha foram agrupados por aba, mas neste caso, foi preciso criar elementos que permitissem selecionar qualquer aba da planilha em cada tabela de destino, e permitissem também, selecionar qualquer coluna da aba selecionada em cada campo da tabela de destino correspondente. Ao final deste processo, todos os campos associados pelo usuário, ou seja, mapeados, são carregados na respectiva tabela na base MAPDB conforme escolhas realizadas.

Validação dos Dados Importados

Após o mapeamento, cada dado importado da planilha, que foi associado a algum campo da base de destino, é verificado. Este processo de validação é muito importante para a ferramenta proposta, pois dados errados não podem ser importados para o sistema Flowup, isso pode ocasionar inconsistências na base Cash.

A coleta dos metadados da base MAPDB é fundamental para validar estes dados da planilha, pois trazem algumas propriedades de todos os campos, de todas as tabelas desta base. Dentre estas propriedades estão:

- IS_NULLABLE - Valor booleano, onde *false* informa que o campo é obrigatório; caso seu valor seja *true* o campo não é obrigatório;
- DATA_TYPE - Indica o tipo do campo (data, decimal, varchar, e outros);
- CHARACTER_MAXIMUM_LENGTH - Indica o tamanho máximo de caracteres suportado pelo campo;

Com isso, cada campo de cada linha da planilha teve seu valor verificado conforme suas propriedades. Por exemplo, se o campo é obrigatório ele não pode ser nulo. No processo, todos os dados são verificados e se ocorrer algum erro, este será adicionado ao relatório de erros. Já outros campos, como código do Banco, CPF/CNPJ, e-mail, fone, dentre outros, precisaram ser verificados de uma forma mais específica.

Há ainda alguns campos que exigiram uma verificação especial. Estes casos são referentes aos campos da base MAPDB que representam um campo da base do Cash que está configurado como chave estrangeira. Nesta condição, o valor verificado da planilha deve ser consultado na tabela referenciada, que corresponde ao campo da chave estrangeira.

Inserção dos Dados na Base Intermediária (MAPDB)

Nesta etapa, os dados mapeados precisam ser carregados na base MAPDB. Isso para que esta base já populada, seja usada na próxima etapa, da migração.

Para implementar isso, os dados mapeados conforme escolha do usuário foram agrupados por tabela. Com isso foi necessário implementar um código para percorrer todas as linhas de uma tabela, campo a campo. Assim, foram verificados todos os campos como descrito na Subseção 3.2.2 e, se todos os campos de uma linha da planilha forem validados, o processo avança para a inserção desta linha na base MAPDB. Caso contrário, os campos invalidados entram no relatório de erros e, conseqüentemente o processo de migração dos dados é interrompido.

3.2.3 Migração dos Dados

O objetivo da etapa da migração foi extrair os dados da base MAPDB, transformá-los e, por último carregá-los na base do Flowup (Cash). Na etapa anterior do mapeamento, a base intermediária, MAPDB, foi populada com os dados da planilha. Agora, na etapa da migração, a base MAPDB foi usada como base de origem na configuração do pacote ETL. Para completar a descrição dos passos na configuração básica do pacote ETL, a base Cash foi configurada como base de destino.

Implementação do Processo ETL

O primeiro passo na implementação do processo ETL foi criar um pacote, onde foram adicionadas as tarefas, ou seja, os DataFlow Tasks. A princípio, foi criado um DataFlow para cada entidade a ser importada. Assim, os componentes necessários para o projeto foram adicionados, configurados e conectados em cada Dataflow criado. Esses componentes podem ser de três tipos: Origem, Transformação e Destino.

No padrão de configuração, estes componentes do DataFlow são configurados com valores fixos, mas neste projeto existe a necessidade de alterar estes valores. E para isso, foram adicionados alguns parâmetros na configuração do pacote.

No decorrer do desenvolvimento da ferramenta, algumas funcionalidades foram adicionadas ao Migrup. E para conseguir implementá-las, foi necessária a criação de

variáveis e a adição de um outro tipo de tarefa ao ControlFlow, chamada de Execute SQL Task.

Os métodos usados para implementar os componentes em cada DataFlow, no processo ETL, foram:

- Origem - Estes componentes têm a função de importar os dados da base MAPDB para alimentar o processo ETL. Então, para cada entidade importada, foi adicionado um componente do tipo Origem. Neste, foi realizada a configuração da conexão com a base de origem, no caso, a base MAPDB e em seguida, foi selecionada uma tabela da base para a coleta dos dados;
- Transformação - Estes componentes têm função de limpar, ajustar, adequar, os dados da tabela configurada no componente de Origem e entregar estes dados transformados aos componentes de Destino ou outros componentes que consigam carregar/atualizar os dados da base Cash, como é o caso do OLE DB Command. Dessa forma, os componentes do tipo Transformação foram adicionados, configurados e conectados aos outros componentes da tarefa;
- Destino - Estes têm a função de receber todos os dados da base MAPDB já transformados e carregar em uma tabela na base Cash. Assim, foi adicionado um componente do tipo Destino e feita a configuração da conexão com a base Cash; depois, foi selecionada uma tabela desta base que corresponde; e por fim, foi necessário realizar o mapeamento entre os campos transformados e a tabela na base do Cash.

Deploy do Pacote ETL

Para possibilitar o *Deploy* do pacote ETL, foi necessário criar um Catálogo no *Integration Services Catalogs* do *SQL Server*. Assim, ao final da implementação do processo ETL, o *Deploy* do pacote para o Catálogo foi realizado após algumas etapas de configuração necessárias neste processo.

Execução do Pacote ETL

O processo ETL implementado só fica acessível pela aplicação web após o *Deploy* para o Catálogo, ou seja, a aplicação web não tem acesso aos parâmetros nem às variáveis do pacote ETL em seu ambiente de implementação. Por isso, foram criadas algumas variáveis no Catálogo, e na sequência estas variáveis foram associadas aos parâmetros do pacote.

3.2.4 Relatório dos Resultados

Na implementação do pacote ETL, foi adicionada uma lógica para contabilizar os registros enviados à base de destino. Esta contagem é realizada durante a execução do processo ETL nas tarefas correspondentes às entidades importadas. Na sequência, os dados resultantes destas contagens foram enviados para uma tabela na base intermediária. Então, ao final da execução do processo, a aplicação web deve realizar uma consulta nesta tabela para carregar os gráficos e tabelas.

3.3 Ferramentas Utilizadas

Várias ferramentas foram utilizadas neste trabalho. Algumas tiveram que ser desenvolvidas, e estão descritas na Subseção 3.3.1. Outras foram instaladas em um computador, para preparar o ambiente de desenvolvimento. Estas estão descritas a seguir:

- Adobe XD - Ferramenta de design de UI/UX para sites, aplicativos web e aplicativos móveis, foi utilizada na construção do protótipo na etapa correspondente do DT;
- MS Visual Studio (VS) 2019 - Ambiente de desenvolvimento integrado (IDE) da Microsoft; é usado no desenvolvimento de software para computador, sites, sistemas web e aplicativos móveis; permite o desenvolvimento em .NET Framework e nas linguagens Visual Basic (VB), C# (C Sharp), dentre outros; este é o ambiente usado na codificação manual do Migrup;
- Internet Information Services (IIS) v10 - Servidor web da Microsoft projetado para o Windows Server. É usado para hospedagem de sites e outros conteúdos na Web; foi usado na criação de um servidor local para a fase de implementação do Migrup;
- SQL Server Management Studio (SSMS) v18.2 - Responsável por gerenciar e administrar todos os componentes do Microsoft SQL Server;
- MS SQL Server Developer 2017 - Sistema da Microsoft responsável pelo gerenciamento de Banco de dados relacional (SGBD); além disso, foi usado para incluir os componentes necessários à solução, como: SSIS, Catálogo SSIS, dentre outros;
- Extensão SSIS no Visual Studio - Biblioteca do SSIS para permitir a implementação de projetos de integração no *Visual Studio*; foi usada para implementar o processo ETL;

- MS SQL Server Integration Services (SSIS) - Plataforma de desenvolvimento de projetos de integração de dados e sistemas de transformações de dados; usado para implantar o projeto ETL do Migrup no *SQL Server*.

Com o ambiente preparado, foi feito o clone do repositório do projeto Flowup e, logo em seguida, foi criado um *branch*. Assim, todas as atualizações realizadas no código implementado no modo *hand coding*, no *Visual Studio*, passaram a ser guardadas no mesmo repositório online do sistema Flowup. No entanto, pelo fato da implementação do processo ETL ocorrer em um outro ambiente usando uma outra linguagem, foi criado um repositório à parte, também online, para guardar todas as atualizações.

3.3.1 Desenvolvimento de Ferramentas Auxiliares

Durante o desenvolvimento da solução, houve a necessidade de criar outras ferramentas e estruturas para ajudar na implementação deste trabalho. O primeiro item desta subseção descreve os métodos usados para a criação da base intermediária (MAPDB), que é fundamental para conectar a etapa de mapeamento (sistema web) com a migração (ferramenta ETL). E no segundo item são apresentados alguns passos da implementação de um subsistema para importar a lista de bancos atualizada do site do Banco Central do Brasil. Esta ferramenta foi usada para possibilitar o uso dos campos relacionados aos bancos (*Nome_Banco* e *Codigo_Banco*) no processo de importação.

Criação de uma Base Intermediária

A criação da base intermediária foi fundamental para o desenvolvimento deste trabalho. Esta subseção descreve os motivos da criação desta base e os passos realizados para sua implementação.

No início da etapa do Desenvolvimento, foi definido que uma ferramenta ETL seria usada na etapa da Migração da ferramenta proposta. Dessa forma, foram realizados alguns testes nesta ferramenta. Assim, foi verificado que a implementação do pacote ETL depende da estrutura de suas bases configuradas, tanto origem como destino. Esta estrutura deve ser totalmente padronizada, ou seja, se um pacote ETL foi configurado para uma tabela de origem, o processo ETL só é executado corretamente se uma tabela com exatamente a mesma estrutura de seus campos for carregada no modo de execução. O mesmo comportamento foi verificado para a base de destino do pacote ETL.

Com isso, a base de origem configurada no pacote ETL não poderia ser a planilha importada; pois, foi definido na etapa do DT, que qualquer planilha Excel poderia ser importada, por isso, não existiria um padrão.

Este problema foi resolvido com a criação de uma base intermediária (*staging area*) no mesmo servidor SQL Server da base de destino. Esta base intermediária, que fica entre a origem (planilha) e o destino (Cash), foi implementada de forma separada da enorme base do sistema Flowup (Cash), nomeada de MAPDB, em referência à etapa do mapeamento.

Esta nova base é de certa forma um espelho da base Cash. Durante o processo geral do Migrup, esta base é populada com os dados da origem após realização do mapeamento, isto é, na etapa do mapeamento, o MAPDB corresponde à base de destino. Já na etapa da migração, esta base foi configurada como base de origem da ferramenta ETL.

Importação da Lista dos Bancos

A importação dos campos relacionados aos bancos *Nome_Banco* e *Codigo_Banco*, presentes em duas tabelas do MAPDB, não poderia ser efetivada sem a atualização da lista dos bancos no sistema do Flowup. Isso, porque para estes campos, ocorre uma consulta na tabela de bancos do Flowup durante o processo de validação, e se o dado não for encontrado, o sistema deve acusar erro. Por isso, foi necessário atualizar a lista de bancos do sistema Flowup com os nomes e códigos dos bancos recentemente criados, como exemplo, *Nubank*.

Por essa necessidade de manter a lista sempre atualizada, foi desenvolvida uma ferramenta de apoio para permitir a importação da lista de bancos atualizada para o sistema Flowup. A lista de bancos é baixada do Site do Banco Central do Brasil que a mantém sempre atualizada. Esta implementação foi realizada em um outro ambiente do sistema Flowup, chamada de *ControlPanel*. Neste, apenas a equipe do Flowup tem acesso; é o mesmo ambiente onde roda a ferramenta de importação legada.

4 Resultados do Processo Design Thinking

Neste capítulo são apresentados os resultados conseguidos das etapas do DT. Cada seção descreve as principais contribuições extraídas para o desenvolvimento deste trabalho em cada etapa do DT.

4.1 Imersão

Para entender bem o problema, foram realizadas entrevistas individuais com alguns membros da equipe do Flowup: *Scrum Master*, Desenvolvedor de software Sênior, um integrante do suporte e outro do marketing, totalizando quatro entrevistas. As perguntas feitas foram sobre o uso da ferramenta legada no processo de importação dos dados de novos clientes. Destas entrevistas, foram extraídas várias informações:

- Existe um formato único de planilha padrão, e os dados precisam ser preenchidos exatamente conforme este formato;
- A planilha é preenchida manualmente, não tem outra opção, então, dependendo do tamanho dos dados, esse tempo pode ser longo;
- Ocorrem vários tipos de erros no preenchimento da planilha: dados incompletos; erro de digitação; dados em um formato diferente do solicitado; ou dados inválidos de CPF, e-mail, telefone, dentre outros;
- A ferramenta valida os campos antes de carregar, mas em caso de falha na importação, não informa nem o local nem a causa do erro, só informa que ocorreu alguma falha no processo;
- Apenas a equipe técnica tem acesso à ferramenta de importação legada.

Usando a ferramenta do DT, pesquisa de soluções similares, foram encontrados alguns sistemas de gestão que possuem uma ferramenta de importação de dados. Os três sistemas encontrados foram testados, sendo cada experiência detalhada da seguinte forma:

- O sistema de gestão online *SIGE Cloud*, permite a importação dos dados através de um arquivo Excel. No site da empresa é explicado que as informações preenchidas na planilha devem ser “*bem registradas e corretamente ordenadas*”, palavras usadas na descrição da funcionalidade. Após realização dos testes, foi

verificado que a importação é realizada corretamente, e uma mensagem é enviada por e-mail informando que a importação foi realizada com sucesso. Porém, foram inseridos valores inválidos de CNPJ, e mesmo assim os registros foram cadastrados. O mesmo ocorreu para os outros campos, por exemplo, e-mail, telefone, CEP, dentre outros (SIGE, 2021);

- A empresa *Omie* também disponibiliza que os próprios usuários realizem as importações das planilhas. Então, o cliente consegue baixar uma planilha pré-formatada para preencher e inserir no sistema. Segundo as informações no site da empresa, o relatório do resultado da migração é enviado ao cliente, por e-mail. E em caso de erro, é informada a linha da planilha que apresentou algum problema. Todas estas informações foram conferidas nos testes realizados. A desvantagem deste sistema de importação é o fato da planilha ser padrão e o cliente precisar preencher manualmente conforme o formato exigido (OMIE, 2021);
- O sistema da empresa *Nibo*, dentre os sistemas encontrados, é o único que permite a importação de qualquer formato de planilha, de certo modo. Pois, em seu sistema de importação, o usuário pode copiar sua planilha do Excel e colar na tela de importação, como mostrado na Figura 17. Ao confirmar a importação, o sistema mostra uma tabela com todos os dados copiados para serem mapeados, onde cada coluna tem uma caixa de seleção com todos os campos existentes da entidade que está sendo importada, conforme Figura 18. Este sistema também permite a atualização dos dados já cadastrados. A desvantagem desse sistema é que ele não valida nenhum campo (NIBO, 2021).

Importar Funcionários

Passo 1: Copie e cole aqui sua lista de contatos do Excel

João	teste1	654987
José	teste2	654654
Maria	teste3	81 - 98563214

Figura 17 – Cópia da tabela do Excel para importação do sistema Nibo
Fonte: Elaborado pelo autor.

Esta etapa foi importante para conhecer melhor o sistema legado e principalmente entender os problemas ocorridos no processo atual de importação dos dados de clientes. Além disso, foi fundamental conhecer as ferramentas de importação de outros sistemas. Assim, vários dados foram levantados, com relação às funcionalidades existentes no mercado e necessidades dos usuários.

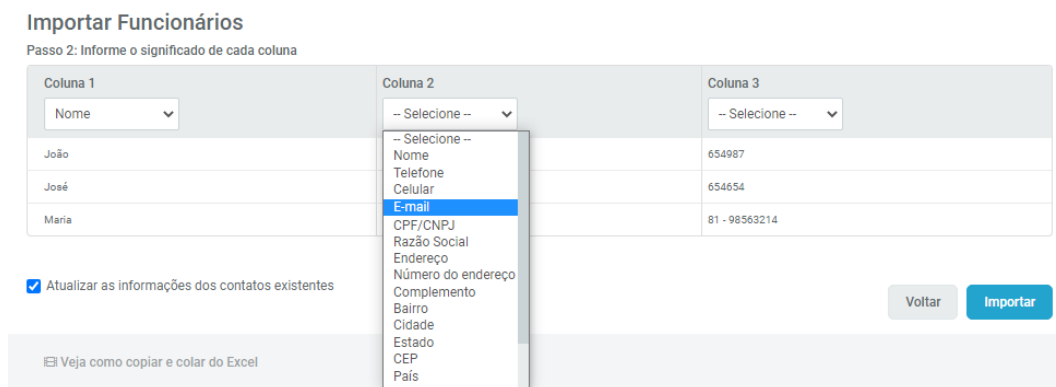


Figura 18 – Mapeamento das colunas com o campo da entidade Fornecedores do sistema Nibo. Fonte: Elaborado pelo autor.

4.2 Análise e Síntese

Essas informações já conseguidas até esta etapa puderam gerar alguns critérios norteadores que serviram de base desde o início até o final do desenvolvimento da nova ferramenta de importação.

AS-IS	TO-BE
Formato único de planilha a ser preenchida manualmente e importada	O preenchimento é manual, mas tem a opção de importar qualquer planilha, inclusive com várias abas
Em caso de erro na planilha, não é informado nem o tipo nem o local	Em caso de erro na planilha, uma tabela com todos os erros é apresentada ao usuário informando a aba e a linha
Apenas a equipe técnica tem acesso	Os clientes (usuários finais) também têm acesso
Não possui a visualização da planilha	Permite a visualização de qualquer aba da planilha
Tem a opção apenas de inserir novos dados dos clientes	Permite inserir novos dados como também, atualizar os dados já cadastrados no sistema

Tabela 1 – Tabela AS-IS X TO-BE da ferramenta de importação dos dados dos clientes do sistema Flowup. Fonte: Elaborado pelo autor.

Para apresentar estes critérios em um formato mais informativo, foi optado por construir um AS-IS/TO-BE, mostrado na Tabela 1. Estes termos, são fases do BPM (*Business Process Management*) para melhoria de processos de negócio. O AS-IS, no contexto deste trabalho, corresponde aos processos atuais do sistema legado; e o TO-BE, representa uma visão dos processos do sistema proposto.

Ao analisar os dados da Tabela 1, podem ser extraídas algumas oportunidades de melhorias totalmente viáveis de serem implementadas neste trabalho. Estas melhorias podem trazer benefícios como: independência dos clientes, flexibilidade ao importar a planilha e agilidade na identificação dos erros desta.

4.3 Ideação

Nesta subseção são apresentadas as ideias compiladas de duas reuniões com dois integrantes da equipe técnica do Flowup, o *Scrum Master* e um Desenvolvedor Sênior. Na primeira reunião foram discutidos os resultados obtidos nas etapas anteriores do DT, usando a técnica de *brainstorm* para tal. Na segunda foram definidos o ambiente onde a ferramenta deve ser implantada e as ferramentas a serem utilizadas.

Uma das decisões mais importantes definidas nesta etapa foi que, nesta primeira versão, a nova ferramenta iria importar apenas planilhas do Excel, e não mais os dados de outros bancos de dados. Esta decisão foi tomada pela falta de padrão das bases de dados de outros sistemas, sendo inviável a construção de uma ferramenta com esta opção.

A escolha de liberar a importação apenas de planilhas não anula a possibilidade de importar os dados de outras bases. Pois, um membro da equipe técnica do Flowup pode acessar o banco de dados do cliente e criar as *Views*, ou seja, visualização de dados específica de uma ou mais tabelas, que resultem nos dados que o cliente deseja importar ao novo sistema. Em seguida, as planilhas poderão ser geradas, usando os dados das *Views*, e importadas pela nova ferramenta.

Algumas ideias puderam ser extraídas, utilizando as informações conseguidas das etapas anteriores e compartilhadas com a equipe do Flowup. As ideias anotadas do ponto de vista de melhor atender às necessidades dos usuários foram:

- A ferramenta proposta deve importar apenas planilhas;
- Estender a importação dos dados para qualquer planilha Excel no formato de tabelas;
- Disponibilizar esta nova ferramenta aos clientes, usuários finais;
- Para facilitar a verificação dos dados importados, pode ser disponibilizada uma pré-visualização das planilhas;
- Implementar a validação dos dados da planilha;
- Durante o processo de validação, todos os erros podem ser registrados e ao final, serem apresentados em uma tabela ao usuário;
- Apresentar um relatório após finalização da migração, informando os dados dos resultados do processo; e caso tenha ocorrido algum erro, apresentar relatório dos erros;
- Adicionar gráficos no relatório da migração mostrando o número de registros importados, como também, a contagem dos campos nulos;

- Adicionar a opção de atualizar os dados já cadastrados dos clientes.

Do ponto de vista da solução técnica, as reuniões realizadas na fase de análise e síntese contribuíram para importantes definições no desenvolvimento da ferramenta proposta, como:

- Implementar a nova ferramenta no mesmo ambiente de desenvolvimento do Flowup, que usa o *Visual Studio* da Microsoft com ASP.NET MVC, e a linguagem C#; isso foi escolhido por utilizar o mesmo ambiente de desenvolvimento e implantação do sistema legado, e assim, facilitando na integração da ferramenta ao sistema Flowup, no conhecimento técnico da equipe, na manutenção e escalabilidade;
- Utilizar uma ferramenta ETL para implementar a importação dos dados da planilha; neste caso foi escolhido o SSIS também da Microsoft; isto foi definido para facilitar na implementação do processo ETL, na manutenção e escalabilidade do Migrup; além disso, a implantação do SSIS não trará nenhum custo adicional pois basta instalar a sua extensão no SQL Server já em uso no sistema Flowup;
- Incluir uma nova base de dados para auxiliar na etapa de migração dos dados, usando a ferramenta ETL; isto foi necessário pelo fato do Migrup trabalhar com qualquer tipo de planilha no formato de tabela, dessa forma não seria viável utilizar uma planilha como origem da ferramenta ETL (SSIS) pois não se sabe como será o formato da planilha que o usuário irá importar.

4.4 Prototipação

O protótipo foi criado nesta etapa do projeto guiado pelas informações já apresentadas, discutidas e validadas. Nesta subseção são apresentadas as telas construídas e as ideias que surgiram após as observações e sugestões anotadas durante a apresentação do protótipo aos membros da equipe do Flowup. Esta etapa ajudou muito no desenvolvimento da ferramenta, pois os detalhes podem ser melhor visualizados.

O protótipo foi feito de forma simples tentando simular apenas as principais funções do Migrup, e pode ser acessado por [Protótipo do Migrup](https://xd.adobe.com/view/93a15017-c0a1-4260-5dc3-a41b09cd2bb2-e590/?fullscreen), ou pela url: <https://xd.adobe.com/view/93a15017-c0a1-4260-5dc3-a41b09cd2bb2-e590/?fullscreen>. Na Figura 19 é mostrada a tela inicial da ferramenta, com o componente para buscar um arquivo Excel no computador e importá-lo para a etapa do mapeamento.

Na sequência, pode ser mostrada a Figura 20, que corresponde à tela de mapeamento entre os campos da base de origem (planilha) e os campos da base de destino.

Migração
Importação da Planilha

A interface de usuário para a importação de uma planilha. No topo, há o título "Importação por arquivo". Abaixo, há um formulário com dois campos de entrada: "Arquivo:" com o placeholder "Escolha o arquivo" e "Caminho completo do arquivo". À direita do formulário, há um botão azul com o texto "Importar".

Figura 19 – Tela inicial do protótipo para importar a planilha.
Fonte: Elaborado pelo autor.

Em cada linha do painel de mapeamento tem a opção de expandir e retrair seus campos. Isto foi interessante para compactar as informações na tela. Ainda nesta figura, os elementos à esquerda são as tabelas e campos da base de destino, e os elementos à direita, correspondem às abas e colunas da planilha importada, que poderão ser selecionados conforme escolha do usuário.

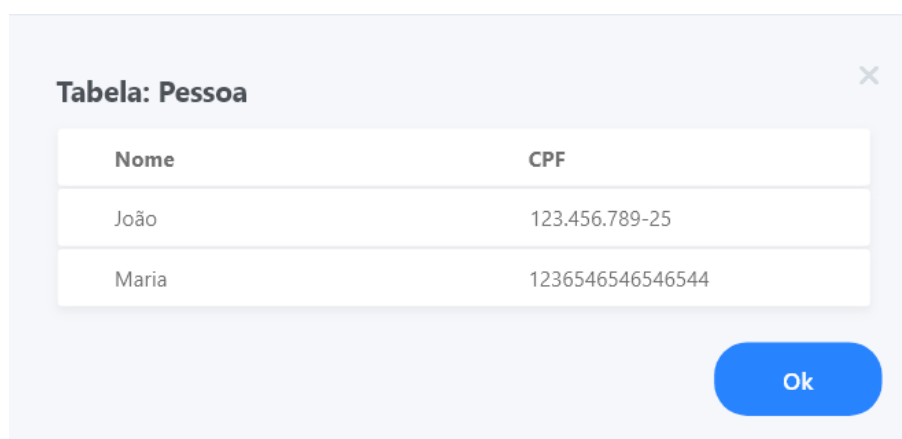
Migração
Mapeamento entre as bases de dados

A interface de usuário para o mapeamento entre bases de dados. Ela é dividida em duas seções principais: "Tabela de Destino" e "Aba da Planilha".
Na seção "Tabela de Destino", há duas tabelas. A primeira tabela, "Nome da Tabela 1 (Ex. Pessoa)", tem uma aba selecionada "Aba Pessoa" e um botão "Visualizar dados". A segunda tabela, "Nome da Tabela 2 (Ex. Empresa)", tem um campo de seleção "Selecione a aba da planilha" e um botão "Visualizar dados".
A seção "Aba da Planilha" contém uma tabela com três colunas: "Nome da coluna", "Campo obrigatório" e "Coluna de origem".
A primeira linha da tabela "Aba da Planilha" tem o nome "Nome da coluna 1 (Ex. Nome)", um campo obrigatório com um asterisco (*) e um campo de seleção "Selecione a coluna da aba".
A segunda linha da tabela "Aba da Planilha" tem o nome "Nome da coluna 2 (Ex. CPF)" e um campo de seleção "Selecione a coluna da aba".
No rodapé da interface, há dois botões: "Cancelar" (cinza) e "Realizar Migração" (azul).

Figura 20 – Tela de mapeamento entre as bases.
Fonte: Elaborado pelo autor.

Os dados da planilha estão representados por caixas de seleção que poderão associar uma tabela de destino com uma aba da planilha. Após selecionar uma aba, é possível visualizar os dados da tabela correspondente através de um botão nomeado de *Visualizar dados* e, então, na Figura 21, pode ser verificada a tabela solicitada, que mostra todos os seus campos e colunas.

Após verificação dos dados da aba, ação esta opcional, as colunas correspondentes à aba selecionada serão carregadas nas caixas de seleção e, assim, poderão



Nome	CPF
João	123.456.789-25
Maria	1236546546546544

Figura 21 – Visualização de uma aba da planilha
Fonte: Elaborado pelo autor.

ser associadas a um campo da tabela de destino, como mostrado na Figura 20.

Ao final do mapeamento, com os campos associados conforme preferência do usuário, este poderá solicitar a execução da migração acionando o botão *Realizar Migração* na Figura 20. Dessa forma há dois fluxos possíveis neste protótipo. Caso todos os dados da planilha associados estejam corretamente preenchidos com dados válidos, a migração será realizada e ao ser concluída deverá aparecer um relatório gráfico informando o número de registros inseridos e atualizados, como apresentado na Figura 22. Porém, se ocorrer alguma falha em algum dado da planilha, com relação às características da coluna associada na tabela de destino, será mostrada uma tabela informando os erros ocorridos durante a tentativa de migrar os dados, Figura 23.



Figura 22 – Relatório do processo de migração dos dados
Fonte: Elaborado pelo autor.

Após apresentação deste protótipo para quatro integrantes da equipe do Flowup e para o orientador, as anotações feitas foram transformadas em algumas ideias para



Aba da planilha	Coluna da planilha	Linha	Mensagem
Pessoa	CPF	2	CPF inválido
Empresa	E-mail	1	Formato inválido

Figura 23 – Tabela de erros encontrados na planilha
Fonte: Elaborado pelo autor.

a ferramenta proposta, porém nem todas foram implementadas nesta primeira versão do Migrup. As ideias anotadas foram:

- Implementar a ferramenta no padrão visual do Cash;
- Além de planilhas do Excel, importar outros tipos de arquivos, csv, json, dentre outros;
- Adicionar um campo na importação da planilha para informar o número da linha que inicia a tabela;
- Na tela de mapeamento, retirar a coluna do Campo Obrigatório e colocar o asterisco ao lado direito dos nomes das colunas obrigatórias;
- Adicionar uma coluna na tela de mapeamento informando o limite de caracteres de cada campo;
- Na tabela de erros, adicionar as colunas referentes à base de destino;
- Adicionar no relatório da migração os gráficos com a contagem de campos nulos importados;
- Mostrar o feedback das etapas do processo de migração em tempo real.

5 Migrup

Uma vez finalizada a etapa de DT, de posse do protótipo e das decisões técnicas, passou-se para a etapa de desenvolvimento da ferramenta. O fluxo que ela implementa, a arquitetura e as telas que ilustram a solução final serão detalhados neste capítulo. As seções foram divididas da seguinte forma: a Seção 5.1 descreve os fluxos do processo; na Seção 5.2 é apresentada a arquitetura da ferramenta proposta; e a Seção 5.3 descreve a solução na visão do usuário.

5.1 Descrição do processo de importação usando o Migrup

Todas as etapas do processo estão descritas nesta seção. Na Figura 24 é apresentado o fluxograma completo do processo. Seguem as descrições de cada etapa:

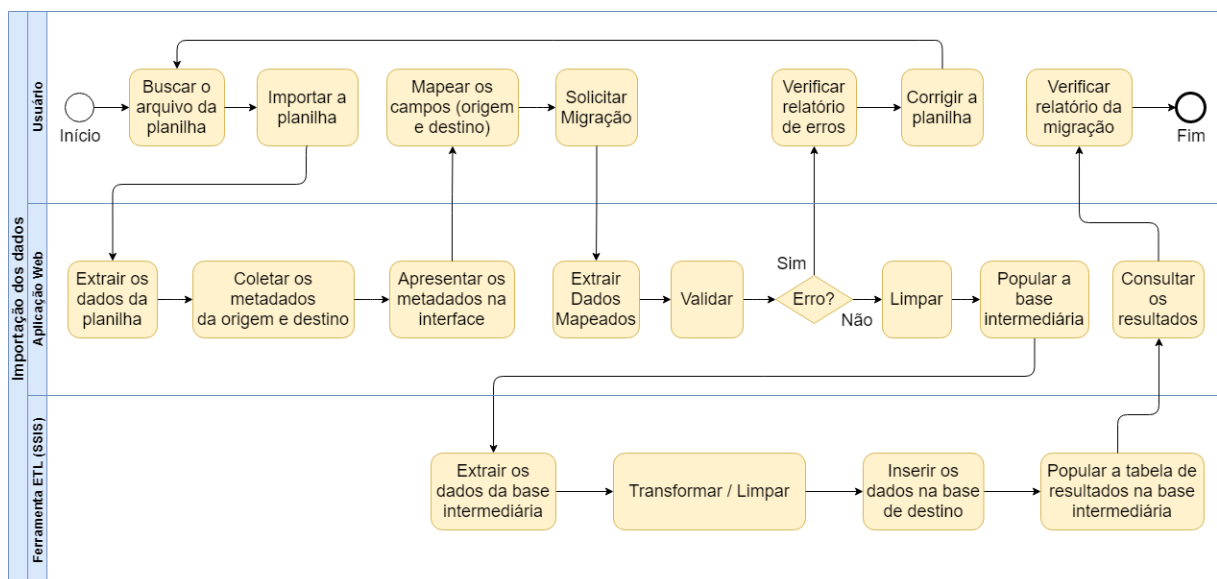


Figura 24 – Fluxograma do processo de importação dos dados usando o Migrup.

Fonte: Elaborado pelo autor.

- Inicialmente, de forma manual, o cliente deve buscar o arquivo da planilha em seu computador, e em seguida, solicitar a importação. Esta planilha pode ser já utilizada pela empresa do cliente, ou pode ter sido preenchida usando a planilha padrão de importação do Migrup;
- A ação de importar a planilha resulta na execução automática de três etapas:
 - Extração dos dados da planilha;

- Coleta dos metadados da planilha (origem) e da base MAPDB (base intermediária);
 - Apresentação de todos os metadados coletados na interface gráfica de forma que o cliente consiga escolher as tabelas que deseja importar para o Flowup;
- O cliente deve, então, utilizar a interface gráfica para escolher as tabelas existentes na planilha importada e associar os campos das tabelas de destino com algum campo de alguma tabela da origem; este mapeamento é realizado de forma manual pelo cliente;
- Ao concluir o mapeamento, o cliente vai solicitar a execução da migração. Esta ação dá início ao restante do processo automático do Migrup. As etapas executadas nesta fase do processo são:
 - Validação dos dados mapeados, de forma detalhada, verificando campo a campo de todas as tabelas selecionadas no mapeamento;
 - Em seguida, é feita a verificação do resultado da etapa da validação; então, se não tiver ocorrido nenhum erro o processo segue para a próxima etapa do fluxo normal, a etapa da limpeza dos dados; porém se algum erro foi detectado em algum dado da planilha, o processo segue para outro fluxo, etapa da verificação dos erros ocorridos. A etapa anterior da validação ocorre de forma que todos os campos de todas as tabelas sejam validadas em vários aspectos, e cada erro encontrado em algum campo é adicionado no relatório de erros;
 - No fluxo normal, sem erros, alguns campos passam por limpeza ou algumas transformações simples, antes de passar para a próxima etapa;
 - Por outro lado, no fluxo que identifica um ou mais erros, aparece a tabela contendo todos os erros detectados na etapa da validação; neste fluxo o processo de migração é interrompido; e então, o cliente deve verificar e corrigir os erros na planilha; depois disso deve reiniciar o processo carregando a planilha corrigida;
 - Voltando à sequência do fluxo normal, após a limpeza, os dados são populados nas suas tabelas correspondentes da base MAPDB conforme associações realizadas na etapa do mapeamento;
- As etapas automáticas, implementadas na ferramenta ETL (SSIS), executadas após a inserção dos dados mapeados na base MAPDB são (mais detalhes sobre os itens abaixo descritos na Seção [A.2.3](#)):
 - A extração dos dados da base MAPDB;

- A limpeza dos dados extraídos usando os componentes de transformação da ferramenta ETL;
 - Inserção dos dados após a limpeza e transformações na base do Flowup (Cash);
- Após a execução da migração, um relatório deve ser apresentado ao cliente informando os números relacionados aos registros inseridos ou atualizados na base de destino; e se ocorrer algum erro, aparecerá a tabela contendo a lista de erros ocorridos no processo.

5.2 Arquitetura do Migrup

Na elaboração da arquitetura, o objetivo principal da fase inicial foi definir quais abordagens seriam usadas na implementação de cada etapa do Migrup. Com isso, foi optado a utilização de uma ferramenta ETL, e pelo fato do Flowup ser implementado com ferramentas da Microsoft, foi mais interessante escolher o SSIS. Essa opção de usar uma ferramenta ETL trouxe alguns benefícios como: facilidade na implementação, e conseqüentemente, na manutenção do Migrup; praticidade no tratamento dos dados; e menor necessidade de conhecimentos mais avançados em banco de dados.

A Figura 25 mostra a arquitetura da ferramenta proposta, com as principais ferramentas utilizadas e o fluxo geral de funcionamento entre cada elemento.

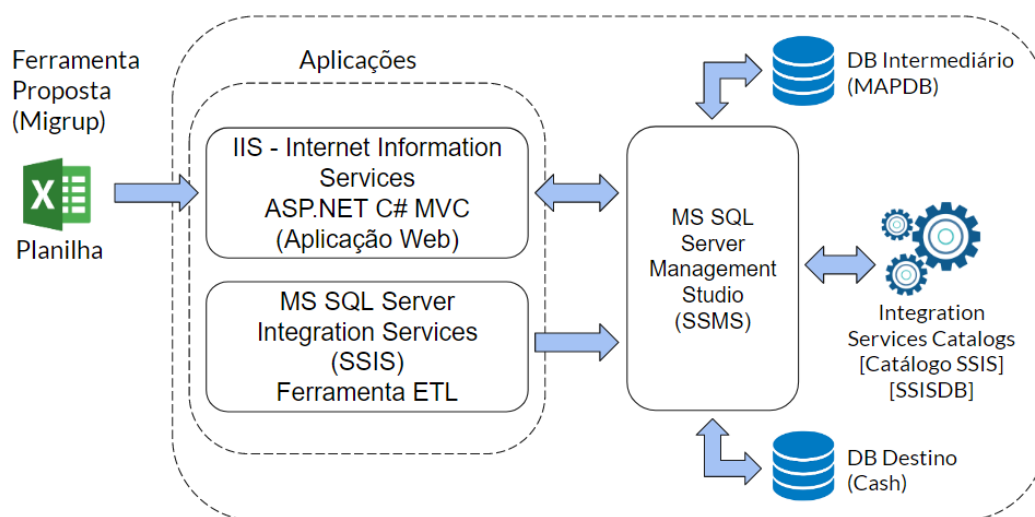


Figura 25 – Arquitetura do Migrup

Fonte: Elaborado pelo autor.

Para utilizar a ferramenta ETL, a base de origem configurada em cada tarefa, correspondendo a cada entidade, deve ter uma estrutura padrão, assim, a base de origem não pode ser uma planilha, pois no contexto deste trabalho, não terá padrão. Por isso, foi necessário criar uma base intermediária que pudesse ter seus campos

fixos e assim poderem alimentar o pacote ETL. Com isso, após o mapeamento dos campos no sistema web, os dados mapeados são gravados na base intermediária, que carrega o processo ETL.

A implementação do processo ETL, utilizado para a etapa da migração das entidades, é realizada na ferramenta ETL (SSIS). Este ambiente é usado apenas na fase de implementação, e ao ser concluída, deve ser feito o Deploy do projeto ou dos pacotes para o SQL Server, mais especificamente para o Catálogo SSIS. A criação deste, juntamente com o SSISDB, deve ser realizada no Integration Services Catalogs do SQL Server, que fica no ambiente de configuração do SSMS. Após o Deploy, o Catálogo SSIS está pronto para ser executado, com o comando vindo da aplicação web.

Podem ser citadas as seguintes ações executadas em cada elemento da arquitetura seguindo o fluxo do processo:

1. A aplicação web coleta os metadados da planilha;
2. A aplicação web coleta os metadados da base MAPDB através do SSMS;
3. Após o mapeamento entre os campos de origem e destino, o processo automático da ferramenta é executado; então, as tabelas mapeadas são populadas na base MAPDB;
4. Na sequência automática do processo, a aplicação web envia o script de execução do pacote ETL para o Catálogo SSIS, através do SSMS;
5. Ao final da execução do pacote ELT, os dados mapeados são gravados na base de destino (Cash); e, os dados com os resultados do processo são gravados em uma tabela da base MAPDB;
6. A aplicação web consulta o relatório da migração na base MAPDB.

5.3 Uso do Migrup

Nesta seção estão descritos todos os resultados do Migrup na visão do cliente, usuários finais.

5.3.1 Coleta dos metadados

O Migrup manteve as mesmas cinco entidades da ferramenta de importação legada do Flowup: Cliente, Fornecedor, Contas bancárias, Receitas/Despesas e Transferências.

A ferramenta proposta permite que apenas um arquivo seja importado por vez. Então, se o cliente tiver os dados de suas entidades em arquivos diferentes, será necessário executar o processo de migração uma vez para cada arquivo. No entanto, o Migrup permite a importação de um arquivo que contenha várias abas, onde cada aba poderá ter os dados de uma entidade. Dessa forma a migração poderá ser executada uma única vez.

Para importar a planilha, como pode ser visto na Figura 26, o cliente deve buscar um arquivo Excel em seu computador e indicar a posição da linha que começa a tabela, isto é, a posição da linha que contém os nomes das colunas de cada tabela.

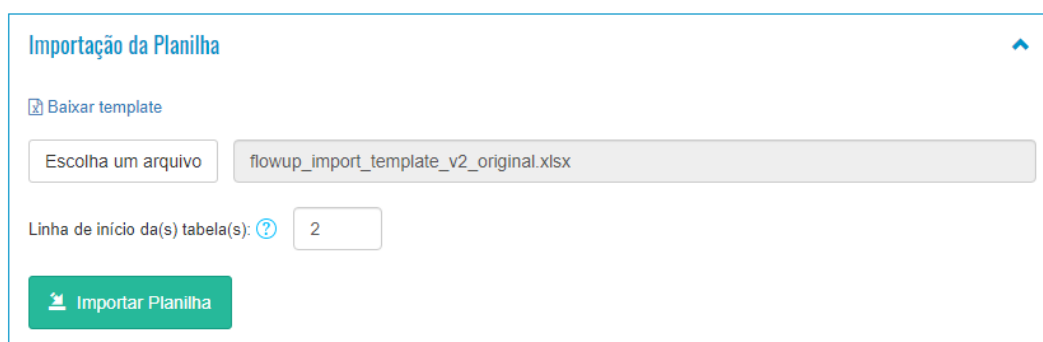


Figura 26 – Painel para escolher o arquivo e importar a(s) planilha(s).
Fonte: Elaborado pelo autor.

5.3.2 Mapeamento entre as tabelas

A etapa do mapeamento foi dividida em duas fases. A primeira foi a associação entre os campos das bases de origem e destino; e, a segunda foi a validação dos dados mapeados.

Associação entres os campos de origem e destino

Nesta fase, todos os metadados coletados foram manipulados para preencher as informações no painel de mapeamento, como pode ser verificado na Figura 27. Nesta, as abas superiores do painel de mapeamento mostram as entidades disponíveis para a importação dos dados, o cliente escolhe qual(is) entidade(s) ele vai querer importar para o sistema Flowup.

Na Figura 27, os campos de destino estão à esquerda e são fixos conforme tabelas da base de destino. Já as campos relacionados à planilha importada estão à direita, incluindo o campo *Aba da Planilha* que fica mais ao centro. Assim, os nomes das abas coletadas foram usadas para preencher as opções do campo *Aba da planilha*. E os nomes das colunas coletadas de cada aba da planilha foram usados para preencher os campos de seleção da coluna à direita do painel do mapeamento *Coluna na Planilha*.

Campo no Flowup	Tamanho Máximo	Coluna na Planilha
Nome_Conta *	100	Nome da Conta
Saldo_Conta *		Saldo da Conta(R\$)
Data_Saldo *		Data do Saldo
Agencia	10	Selecione ...
Conta_Corrente	20	Selecione ...
Codigo_Banco *	5	Selecione ... Selecione ... Nome da Conta Código do Banco Nome do Banco Saldo da Conta(R\$) Data do Saldo Agência Conta Corrente
Nome_Banco	100	

Figura 27 – Painel de mapeamento mostrando a tabela Conta_Bancaria
Fonte: Elaborado pelo autor.

Então, ao selecionar uma aba no campo *Aba da planilha*, todos os campos de seleção da *Coluna na Planilha* mostram as colunas correspondentes da aba selecionada em suas opções. Neste momento, o usuário pode associar cada campo de destino (à esquerda) com algum campo da aba da planilha (à direita). Além disso é possível verificar os dados da aba selecionada no botão *Visualizar dados*.

Validação dos dados importados

Nesta fase, todos os dados da planilha são verificados antes de tentar importá-los ao sistema Flowup. Esta verificação é feita quanto ao formato, validade e propriedades referentes ao campo no Flowup. Os dados da planilha importados são verificados da seguinte forma:

- Campo obrigatório no Flowup não podem ser nulo;
- A quantidade de caracteres na coluna não deve ultrapassar o limite do campo associado no Flowup;
- Os campos de Data devem estar em um formato compatível;
- Os campos de Telefone precisam ter a quantidade de dígitos que correspondam a um telefone fixo (10 dígitos) ou um celular (11 dígitos), já contando com os dois dígitos do DDD da cidade;
- Os campos de e-mail precisam estar em um formato compatível;

- Os campos de CPF/CNPJ precisam ser válidos; o formato é corrigido, então basta que os dígitos preenchidos sejam validados;
- Para alguns campos no Flowup é necessário verificar se o valor da planilha já existe no sistema. São campos como: `Codigo_Banco` na tabela `Conta_Bancária`, `Nome_Conta` na tabela `Receitas_Despesas`, `Empresa_Origem` na tabela `Transferencia`, dentre outros.

Ao final do processo de validação, se algum dado errado for encontrado, é exibida uma tabela com todos os erros, como mostrado na Figura 28. Nesta figura, as duas primeiras colunas estão relacionadas à base de destino, Tabela no Flowup e Campo no Flowup; as quatro seguintes, correspondem às colunas da planilha, Aba da Planilha, Coluna na Planilha, Linha e Valor; e por fim, a última coluna corresponde à mensagem de erro. Assim, pode ser verificado em cada linha a localização na planilha, incluindo o número da linha que ocorreu o erro e a mensagem informando o tipo.

Em seguida o usuário deve corrigir todos os erros encontrados e reiniciar o processo de importação da planilha com os dados já corrigidos.



Tabela no Flowup	Campo no Flowup	Aba da Planilha	Coluna na Planilha	Linha	Valor	Mensagem
Cliente	CPF_CNPJ	Cientes	Cpf/Cnpj	3	047.528.800-95	CPF/CNPJ Inválido
Cliente	CPF_CNPJ	Cientes	Cpf/Cnpj	4	85.849.400/0001-20	CPF/CNPJ Inválido
Fornecedor	Email	Fornecedores	E-mail	3	email2@gmail.com	Formato de Email Inválido
Fornecedor	Nome_Fornecedor	Fornecedores	Nome	4		Campo obrigatório está com valor nulo!
Conta_Bancaria	Data_Saldo	Contas	Data do Saldo	7	12/09/2021a	Valor de data inválida na planilha!
Conta_Bancaria	Codigo_Banco	Contas	Código do Banco	8	3415	Valor não encontrado no sistema.

Figura 28 – Exemplo de dados invalidados apresentados na tabela de erros

Fonte: Elaborado pelo autor.

5.3.3 Relatório dos resultados

Na última etapa da importação, o relatório com os resultados do processo de migração dos dados é apresentado ao usuário. Este relatório em forma de gráficos e tabelas aparecem de acordo com os dados das entidades importadas. Na versão atual do Migrup, podem aparecer um total de cinco gráficos, cada uma com sua tabela. Estes gráficos foram divididos em duas figuras, que estão explicadas a seguir.

Na Figura 29, o primeiro gráfico, à esquerda, representa a soma de todos os registros, das entidades importadas, divididos entre inseridos e atualizados; o segundo

gráfico mostra os registros inseridos, separados por entidade; e, o terceiro corresponde aos registros atualizados, que também são separados por entidade.

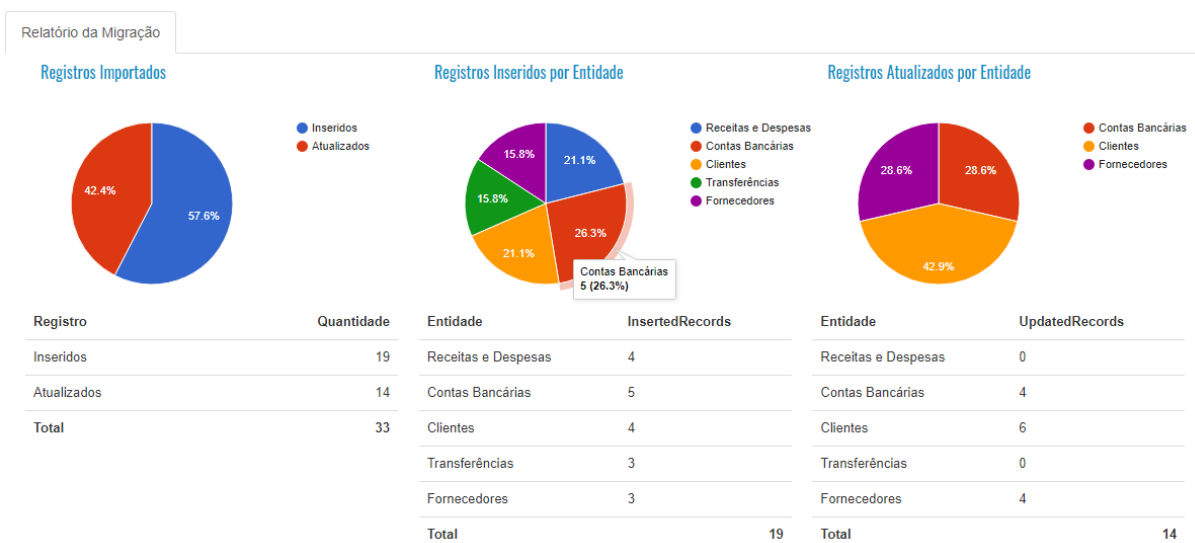


Figura 29 – Experimento 1: Gráficos apresentados no relatório da migração
 Fonte: Elaborado pelo autor.

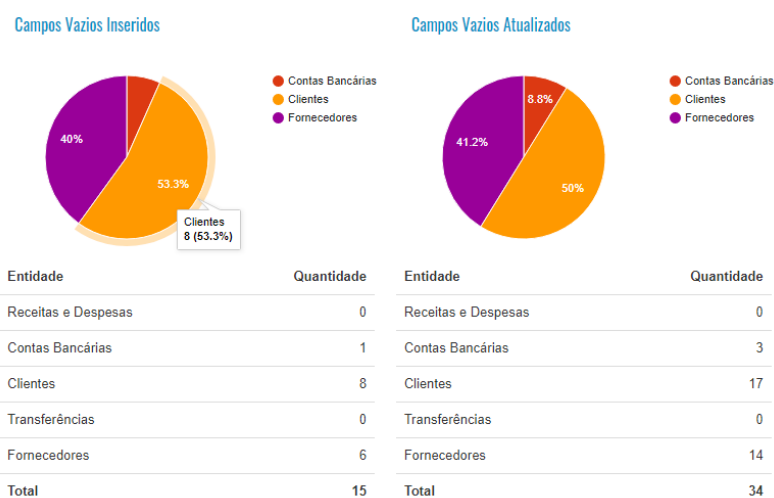


Figura 30 – Experimento 1: Gráficos apresentados no relatório da migração
 Fonte: Elaborado pelo autor.

A segunda parte do relatório, mostrado na Figura 30, apresenta os dois outros gráficos junto com suas tabelas: o quarto gráfico mostra a quantidade de campos vazios inseridos, separados por entidade; e, o quinto, mostra os campos vazios atualizados, separados também por entidade.

Os resultados de outro experimento de migração de dados pode ser verificado nas Figuras 31 e 32. Neste segundo experimento, apenas as entidades Clientes e Fornecedores foram importadas. Por isso, alguns campos aparecem com valor zero nas tabelas.

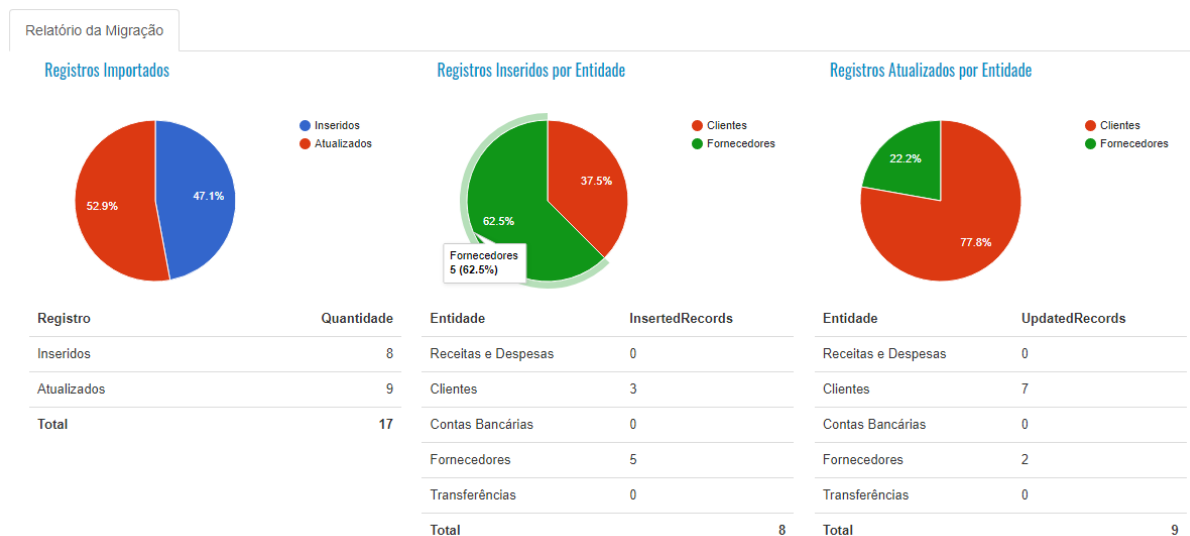


Figura 31 – Experimento 2: Gráficos apresentados no relatório da migração
 Fonte: Elaborado pelo autor.

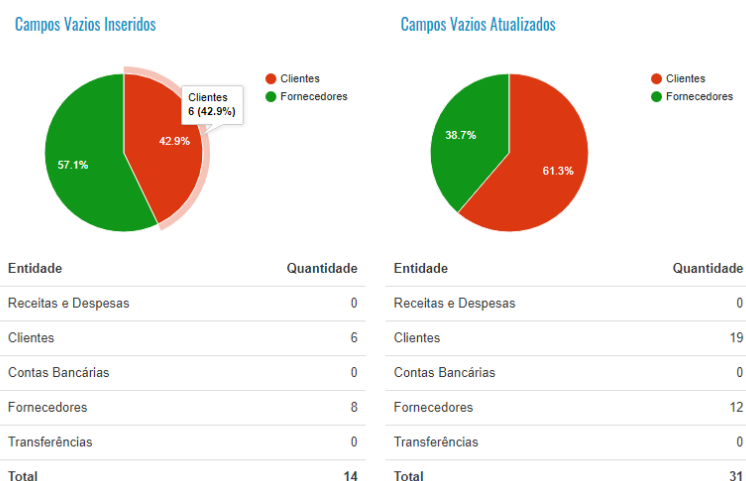


Figura 32 – Experimento 2: Gráficos apresentados no relatório da migração
 Fonte: Elaborado pelo autor.

5.4 Discussão

Esta seção descreve o *feedback* após apresentação das funcionalidades desenvolvidas no Migrup à equipe do Flowup. Além disso, foi realizado um comparativo das ferramentas de importação dos dados de clientes dos sistemas mencionados neste trabalho. Então, as principais funcionalidades e características das ferramentas destes sistemas foram comparadas com as do Migrup.

As funcionalidades do Migrup foram apresentadas para cada membro da equipe do Flowup. O *feedback* foi bem satisfatório, com poucas solicitações de ajuste. Segundo os técnicos, pessoal do suporte e marketing, que realizam as importações atualmente usando a ferramenta legada, em questão de funcionalidades, o Migrup conse-

guiu suprir as principais necessidades. Esta mesma opinião foi compartilhada pelo Scrum Master, mas, por outro lado, sugeriu uma melhoria no visual dos elementos implementados, como botões, tabelas e painéis.

Funcionalidades / Características	SIGE Cloud	Omie	Nibo	Ferramenta Legada	Migrup
Disponibiliza que o usuário importe seus dados	✓	✓	✓	-	✓
Permite apenas a importação dos dados de uma planilha padrão pré-formatada	✓	✓	-	✓	-
Permite a importação de qualquer formato de planilha	-	-	✓	-	✓
Disponibiliza o mapeamento entre os campos de destino com as colunas da planilha	-	-	✓	-	✓
Permite a importação de uma planilha com qualquer número de abas	-	-	-	-	✓
Valida os dados antes de carregar no sistema	-	✓	-	✓	✓
Corrige o formato de alguns dados	-	-	-	-	✓
Feedback da validação dos dados informando o local do erro na planilha	-	✓	-	-	✓
Permite a atualização dos dados já cadastrados no sistema	-	-	✓	-	✓
Feedback do processo realizado com sucesso	✓	✓	✓	✓	✓
Relatório com os dados registrados no sistema	-	-	-	-	✓
Notificações por e-mail	✓	✓	-	-	-

Tabela 2 – Comparativo das ferramentas de importação dos dados de clientes, fornecidas pelos sistemas de gestão citados neste trabalho, com o Migrup.

Fonte: Elaborado pelo autor.

Na tabela 2 estão incluídos os sistemas, fornecedores das ferramentas, que foram citados neste trabalho, além da ferramenta legada e da ferramenta proposta. Então, foram incluídos: os três sistemas pesquisados, SIGE Cloud, Omie e Nibo; mais, a ferramenta legada do sistema Flowup; e, o Migrup. Na primeira coluna, à esquerda, foram inseridas as principais funcionalidades e características das ferramentas de cada sistema, como também do Migrup. As outras colunas correspondem às ferramentas de importação dos sistemas, e na última, o Migrup.

Dessa forma, cada campo destas outras colunas da tabela 2 foi preenchido ou com o símbolo “✓”, indicando que a ferramenta possui a funcionalidade da linha

correspondente, ou com o símbolo “-”, este indicando que a ferramenta não possui a funcionalidade da linha correspondente.

Analisando os dados da tabela 2, foi verificado que o Migrup inclui quase todas as funcionalidades ou características das outras ferramentas. Nesta condição, as funcionalidades implementadas no Migrup podem trazer vários benefícios aos usuários do sistema Flowup conforme análise.

6 Conclusão

Este trabalho apresentou o desenvolvimento e implementação de uma ferramenta, nomeada de Migrup, para a importação dos dados de clientes do sistema Flowup. Dentre as três ferramentas pesquisadas (*SIGE Cloud*, *Omie* e *Nibo*) com o mesmo propósito no mercado nacional, o trabalho proposto inclui funcionalidades que não existem em nenhuma delas e que são importantes do ponto de vista dos entrevistados. Até foram feitas algumas pesquisas por ferramentas estrangeiras mas que não foi possível realizar o cadastro de forma gratuita, para realizar os testes.

A metodologia utilizada na fase de desenvolvimento foi o *Design Thinking*, que ajudou bastante na criação de mais ideias para a ferramenta e aderência às reais necessidades dos usuários. Na fase da implantação, estas novas ideias se transformaram em funcionalidades.

O Migrup oferece vários benefícios ao processo de importação dos dados de clientes novos ou antigos. Os principais benefícios percebidos foram:

- Maior flexibilidade na importação da planilha com a possibilidade de importar qualquer planilha com formato de tabela, e com qualquer número de abas;
- Mais agilidade no processo de importar a planilha, pois nos casos onde há erros, é informado o local exato;
- Correção do formato de alguns dados como data e CPF/CNPJ;
- A ferramenta foi implantada como uma funcionalidade do sistema Flowup, desse modo, permitindo que os clientes possam acessar;
- Inclusão do feedback do processo geral de migração dos dados, através de gráficos e tabelas.

Além destes benefícios oferecidos aos clientes, o método utilizado na etapa da migração dos dados ao destino, do processo geral, pode facilitar na manutenção e escalabilidade do Migrup, principalmente por causa do uso de uma ferramenta ETL.

Para trabalhos futuros, vai ser necessário realizar uma avaliação geral do Migrup pensando agora nos aspectos de tempo de resposta, segurança e implantação da nova funcionalidade no mesmo servidor do Flowup. Além disso, pode ser elaborado um questionário direcionado aos clientes do Flowup, para que o resultado da solução proposta seja validada por mais pessoas e principalmente pelos usuários finais do sistema.

Além disso, com as informações conseguidas nas etapas do *Design Thinking* e Desenvolvimento, foram geradas algumas ideias que não foram implementadas neste trabalho, mas que podem se tornar melhorias em uma nova versão do Migrup. As futuras melhorias identificadas foram:

- Além do XLS e XLSX, verificar a possibilidade de importar outros formatos como: CSV, JSON, TXT, dentre outros;
- Melhorar o visual da tela de importação;
- Permitir a importação de outras entidades, como por exemplo: Projetos, Quadros e Tarefas;
- Feedback das etapas do processo de migração em tempo real;
- Explorar melhor os dados do SSISDB, que possui os dados de execução dos pacotes ETL, entre outras informações;
- Gerar planilhas a partir de Views específicas criadas no servidor de banco de dados de um determinado cliente, para importar os dados de outros sistemas de gestão;
- Alterar nome da base intermediária (MAPDB) para se adequar melhor às suas finalidades.

Para outras possibilidades futuras, vale mencionar, no contexto deste trabalho, que o sistema Flowup pode deixar de usar um banco de dados relacional (SQL), para usar um banco de dados *NoSQL*, que podem trabalhar com bancos de dados relacionais e não relacionais. E para este último caso, o processo usado na construção do Migrup poderia ser também o ELT descrito no Item [2.4.1](#) do referencial teórico.

Referências

- ALI, A. A.; MOHAMED, W. M. Using a business activity monitoring and soa for a real-time etl. *Int. J. Comput. Appl*, v. 180, n. 10, p. 975–8887, 2018. Citado na página 22.
- BROWN, T. Design thinking. *Harvard business review*, v. 86, n. 6, p. 84, 2008. Citado na página 18.
- CHEN, C.-S.; LIANG, W.-Y.; HSU, H.-Y. A cloud computing platform for erp applications. *Applied Soft Computing*, Elsevier, v. 27, p. 127–136, 2015. Citado na página 13.
- COLONETTI, E. A. C. Sistema integrado de gestão empresarial: as vantagens com a troca de fornecedor. *Gerência de Projetos de Tecnologia da Informação-Unisul Virtual*, 2017. Citado na página 14.
- DAS, A. *Introduction to the SSIS Catalog database (SSISDB)*. 2020. Disponível em: <<https://www.sqlshack.com/author/aveek-das/>>. Citado na página 32.
- EL-SAPPAGH, S. H. A.; HENDAWI, A. M. A.; BASTAWISSY, A. H. E. A proposed model for data warehouse etl processes. ELSEVIER, 2017. Citado 2 vezes nas páginas 21 e 22.
- ELMONEM, M. A. A.; NASR, E. S.; GEITH, M. H. Benefits and challenges of cloud erp systems—a systematic literature review. *Future Computing and Informatics Journal*, Elsevier, v. 1, n. 1-2, p. 1–9, 2016. Citado 2 vezes nas páginas 13 e 14.
- FAST. *Sistema Flowup*. 2021. Disponível em: <<https://flowup.me/>>. Citado na página 16.
- FERRO, D. A.; FERREIRA, M. N. A importância do sistema integrado de gestão empresarial para as instituições privadas ou públicas. <https://www.webartigos.com/artigos/a-importancia-do-sistema-integrado-de-gestao-empresarial-para-as-instituicoes-privadas-ou-publicas/112182>, v. 87, p. C3, 1999. Citado 2 vezes nas páginas 13 e 14.
- FEW, S.; EDGE, P. Dashboard confusion revisited. *Perceptual Edge*, p. 1–6, 2007. Citado na página 16.
- HUSSAIN, S. *ETL vs ELT Features and Use Cases*. 2019. Disponível em: <<https://www.mssqltips.com/sqlservertip/5937/etl-vs-elt-features-and-use-cases/>>. Citado 2 vezes nas páginas 34 e 35.
- JAVAPPOINT. *Difference between ETL and ELT*. 2021. Disponível em: <<https://www.javatpoint.com/difference-between-etl-and-elt>>. Citado na página 35.
- KIMBALL, R.; CASERTA, J. *The data warehouse ETL toolkit*. [S.l.]: John Wiley & Sons, 2004. Citado 3 vezes nas páginas 20, 22 e 23.

KNIGHT, B.; KNIGHT, D.; MOSS, J. M.; DAVIS, M.; ROCK, C. *Professional Microsoft SQL Server 2014 Integration Services*. [S.l.]: John Wiley & Sons, 2014. Citado 8 vezes nas páginas 24, 26, 27, 28, 29, 30, 31 e 34.

MALI, N.; BOJEWAR, S. A survey of etl tools. *International Journal of Computer Techniques*, Oct, v. 2, n. 5, p. 20–27, 2015. Citado 2 vezes nas páginas 20 e 21.

MEGIDO, V. F. *A revolução do design: Conexões para o século XXI*. [S.l.]: Editora Gente Liv e Edit Ltd, 2017. Citado na página 18.

MICROSOFT. *catalog.set_environment_variable_value (Banco de Dados SSISDB)*. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/integration-services/system-stored-procedures/catalog-set-environment-variable-value-ssisdb-database?view=sql-server-ver15>>. Citado na página 34.

NIBO. *Ferramenta de importação de dados da Nibo*. 2021. Disponível em: <<https://www.nibo.com.br/empresa/funcionalidades/importacao-de-dados/>>. Citado na página 49.

OLIVEIRA, C. d.; MARCELINO, M. Metodologias e extratécnicas de migração de dados. *Sinergia (CEFETSP)*, v. 13, n. 3, p. 183–191, 2012. Citado na página 14.

OMIE. *Ferramenta de importação de dados da Omie*. 2021. Disponível em: <<https://ajuda.omie.com.br/pt-BR/articles/1439022-importacao-de-dados-em-massa-por-planilha>>. Citado na página 49.

PINHEIRO, T.; ALT, L. *Design Thinking Brasil: empatia, colaboração e experimentação para pessoas, negócios e sociedade*. [S.l.]: Alta Books Editora, 2018. Citado na página 18.

RODRIGUES, F.; COLES, M.; DYE, D. *Pro SQL Server 2012 Integration Services*. [S.l.]: Apress, 2012. Citado na página 23.

ROSS, M.; KIMBALL, R. *The data warehouse toolkit: the definitive guide to dimensional modeling*. [S.l.]: Wiley, 2013. Citado na página 20.

SIGE, C. *Ferramenta de importação de dados da SIGE Cloud*. 2021. Disponível em: <<https://blog.sigecloud.com.br/migrar-dados-planilhas-excel-para-erp/>>. Citado na página 49.

SILVA, L. G.; PESSOA, M. S. d. P. *Uma visão dos sistemas erp*. São Paulo: UNIP, 2004. Citado na página 13.

TOK, W.-H.; PARIDA, R.; MASSON, M.; DING, X. *Microsoft SQL Server 2012 Integration Services*. [S.l.]: Pearson Education, 2012. Citado 2 vezes nas páginas 31 e 32.

VERBEECK, K. *SQL Server Integration Services 2016 Incremental Package Deployment*. 2015. Disponível em: <<https://www.mssqltips.com/sqlservertip/3676/sql-server-integration-services-2016-incremental-package-deployment/>>. Citado na página 33.

VIANNA, M. *Design thinking: inovação em negócios*. [S.l.]: Design Thinking, 2012. Citado 4 vezes nas páginas 17, 18, 19 e 20.

WU, T. Etl function realization of data warehouse system based on ssis platform. In: . [S.l.: s.n.], 2010. p. 1–4. Citado 2 vezes nas páginas 20 e 21.

A Apêndice

A.1 Ferramentas utilizadas na implementação

Nesta subseção são descritas as principais características e funções das ferramentas utilizadas na implementação do Migrup.

A.1.1 MS Visual Studio (VS)

O *Visual Studio* foi utilizado para implementar as etapas de coleta dos metadados e mapeamento dos campos. Neste mesmo ambiente, estão implementadas todas as outras funcionalidades do Flowup, citadas na Seção 2.1. Assim, foi criada a estrutura da nova funcionalidade no mesmo projeto de arquitetura de software Modelo-Visão-Control (MVC) do Flowup. A linguagem de programação usada neste ambiente ASP.NET é C#.

A.1.2 MS SQL Server Integration Services (SSIS)

A implementação do processo ETL foi realizada na extensão SSIS do Visual Studio. Este ambiente de desenvolvimento gráfico permite que o pacote ETL seja implementado de forma mais rápida e prática, utilizando vários componentes e funções para facilitar. Após o *Deploy* para o Catálogo SSIS, todas as extrações, transformações e cargas implementadas no SSIS são implantadas no SQL Server.

A.1.3 MS SQL Server Management Studio (SSMS)

Fora a importação da planilha, todos os outros processos implementados neste trabalho são gerenciados pelo SSMS. Na Figura 25 são indicadas todas as conexões entre as ferramentas usadas neste trabalho. E praticamente no centro de controle está o SSMS, gerenciando quase tudo.

A.2 Detalhes de implementação do Migrup

Nesta seção estão descritos os detalhes da implementação de cada etapa da ferramenta desenvolvida neste trabalho.

A.2.1 Coleta dos metadados

Esta subseção descreve como foi realizada a implementação da coleta dos metadados da planilha e da base intermediária.

Planilha eletrônica

A coleta dos metadados da planilha é realizada quando o usuário pressiona o botão *Importar Planilha*, Figura 26. A ferramenta, então, verifica primeiramente o formato do arquivo e retornar uma mensagem caso não seja um arquivo válido, Figura 33. Em seguida, se o arquivo importado for validado, o código mostrado na Figura 34 é executado uma vez para cada aba da planilha importada. Dessa forma, apenas a primeira linha da tabela é extraída. Ao final deste processo, os nomes das colunas das tabelas são separadas pelo nome da aba e guardadas em listas, que foram usadas para preencher os campos relacionados à planilha no ambiente de mapeamento.

Formato não aceito, arquivo inválido ou número da linha digitado incorretamente! Os formatos aceitos são .XLS e .XLSX



Figura 33 – Mensagem informando que o arquivo importado não foi validado
Fonte: Elaborado pelo autor.

```
while (true)
{
    string columnName = GetCellValueAsString(row.GetCell(columnNumber));

    if (string.IsNullOrEmpty(columnName))
    {
        break;
    }
    else
    {
        listColumnNames.Add(new DataItemMigration
        {
            Id = columnNumber.ToString(),
            Name = columnName,
        });
    }
    columnNumber += 1;
}
```

Figura 34 – Código para coleta dos metadados do arquivo Excel
Fonte: Elaborado pelo autor.

Base intermediária

A coleta dos metadados da base intermediária, MAPDB, é executada após a solicitação de importação da planilha, realiza pelo usuário, assim como é feito para a coleta dos metadados da planilha.

Na Figura 35 é mostrado o *script* chamado para coletar os metadados de uma tabela do esquema *dbo* na base MAPDB. Este *script* é executado uma vez para cada

tabela, e ao final, de forma semelhante como ocorre na coleta dos metadados da planilha, os metadados da base MAPDB são guardados em listas separadas pelos nomes das tabelas existentes no esquema *dbo*.

```
string queryString = string.Format(@"SELECT ORDINAL_POSITION, COLUMN_NAME, IS_NULLABLE, CHARACTER_MAXIMUM_LENGTH, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_SCHEMA = 'dbo'
AND TABLE_NAME = '{0}';", tableName);

var connectionString = WebConfigurationManager.ConnectionStrings["MAPDBContext"].ConnectionString;
```

Figura 35 – *script* para coleta dos metadados da base MAPDB
Fonte: Elaborado pelo autor.

A.2.2 Mapeamento entre as tabelas

A etapa do mapeamento foi dividida em três fases. A primeira foi a associação entre os campos das bases de origem e destino; a segunda foi a validação dos dados mapeados e, a terceira foi a inserção destes dados na base MAPDB. Nesta subseção são descritos os detalhes da implementação destas fases do mapeamento.

Associação entres os campos de origem e destino

Nesta fase, os metadados extraídos da base MAPDB preenchem os elementos do painel de mapeamento mostrados à esquerda na Figura 27. Estes elementos servem de referência para que o usuário possa associá-los aos campos da aba planilha selecionada. Além disso, eles representam os dados da tabela de destino do Flowup, Cash, e são fixos. São alterados apenas se houver alguma mudança na base do Cash, que exigiria a mesma alteração na base MAPDB. No caso da planilha Excel, os metadados foram usados nas opções das caixas de seleção para serem manipulados pelo usuário.

Validação dos dados importados

Inicialmente, todos os campos são verificados de acordo com as propriedades coletadas da base MAPDB. O Fluxograma geral da implementação desta fase pode ser verificado na Figura 36. Então a princípio, os dados são validados quanto às suas propriedades, que são três: IS_NULLABLE, DATA_TYPE e CHARACTER_MAXIMUM_LENGTH, assim, se houver alguma incompatibilidade dos campos mapeados da planilha com referência ao campo da tabela na base MAPDB, o erro é adicionado a uma lista.

Após essa verificação inicial, alguns campos ainda são verificados de forma mais específica. Então, campos como telefone, e-mail, CPF e CNPJ são verificados por funções específicas no código, e no caso dos dados inválidos, todos são adicionados na lista de erros. Para os casos do CPF e CNPJ, seu valores são limpos e

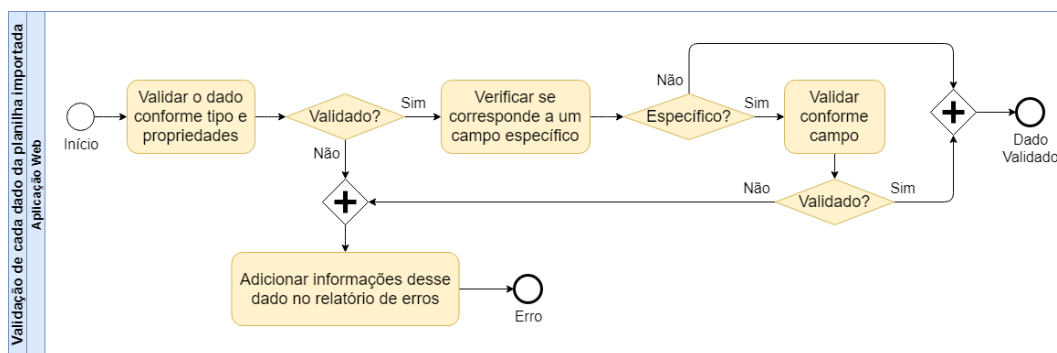


Figura 36 – Fluxograma geral da implementação das validações, correspondendo a cada dado importado. Fonte: Elaborado pelo autor.

convertidos para números e, depois, verificados com seus respectivos algoritmos de validação. Para estes dois dados do Flowup, o mesmo campo pode ser preenchido por um ou outro. Deste modo, além de validar os números conforme dados importados da planilha, é identificado também se corresponde a um CPF ou CNPJ. Dessa forma, um outro campo da base de destino que informa o tipo do Cliente ou Fornecedor é preenchido com “Física” ou “Jurídica”.

Há ainda alguns campos que, para serem validados, dependem do resultado de uma consulta SQL a uma tabela na base Cash (mais detalhes na Subseção A.3.1). Estes são os casos onde o campo da base MAPDB corresponde a uma *chave estrangeira* (*foreign key*) na base Cash, como por exemplo, o campo *Apelido_Cliente* da tabela *Receita_Despesa* da base MAPDB representa o campo *Client_Id* da tabela *CashFlowItems* da base Cash, mostrado na Figura 37. Logo, usando este exemplo, para validar um dado da coluna *Apelido_Cliente* da tabela *Receita_Despesa*, deve ser realizada uma consulta na tabela referenciada, *Clients*, na base Cash. Com isso, se o valor do dado consultado não for encontrado na tabela referenciada, as informações são adicionadas na lista de erros.

CashFlowItems			
	Column Name	Data Type	Nullable
🔑	Id	int	No
	Date	datetime	Yes
	Description	nvarchar(200)	No
	Value	decimal(18, 2)	No
	Generated	bit	No
	Executed	bit	No
	TransactionId	nvarchar(MAX)	Yes
	Timestamp	datetime	No
	BankAccount_Id	int	No
	Category_Id	int	No
	Parent_Id	int	Yes
	Client_Id	int	Yes
	Company_Id	int	No

Receita_Despesa			
	Column Name	Data Type	Nullable
🔑	Id	int	No
	Data_Vencimento	datetime	No
	Descricao	varchar(200)	No
	Nome_Conta	varchar(100)	No
	Projeto_Depto	varchar(100)	Yes
	Categoria	varchar(100)	No
	Empresa	varchar(100)	No
	Apelido_Cliente	varchar(60)	Yes
	Nome_Fornecedor	varchar(100)	Yes
	Colaborador	varchar(100)	Yes
	Numero_NF	varchar(200)	Yes
	Data_Compensacao	datetime	Yes
	Data_Compentencia	datetime	Yes

Figura 37 – Chaves estrangeiras em alguns campos da base Cash, à esquerda, representadas pelos campos correspondentes da tabela no MAPDB, à direita.

Fonte: Elaborado pelo autor.

Inserção dos dados na base intermediária

Nesta fase do mapeamento, os dados validados são inseridos na base MAPDB. A função de inserção, implementada neste trabalho, utiliza o mesmo ciclo para validar e inserir os dados de uma linha da aba da planilha, conforme resultado das validações. Esta função é executada uma vez para cada aba da planilha, sendo cada linha da aba, percorrida a cada ciclo. Portanto, se todos os dados forem validados o comando de inserção é executado.

A.2.3 Implementação do pacote ETL

Neste item são apresentados os resultados da implementação da etapa da migração, realizada na ferramenta ETL. No Item *ControlFlow* são descritas as implementações de forma geral em todo pacote ETL. Os outros itens descrevem as implementações realizadas nos componentes do *DataFlow Task* utilizados neste trabalho. Estes componentes são divididos em três tipos: Origem, Transformação e Destino.

ControlFlow

Ao criar o pacote ETL, as tarefas foram adicionadas e implementadas no *ControlFlow* conforme sua finalidade. Para isto, foi preciso seguir os seguintes passos: inserir os componentes necessários, realizar as conexões entres eles e, configurar cada um de acordo com seu propósito.

A configuração padrão dos componentes das tarefas do tipo *DataFlow Task* pode ser alterada apenas no modo de desenvolvimento. Então, para permitir que esta configuração seja alterada no modo de execução, foi necessário adicionar expressões específicas em alguns componentes. A configuração destas expressões, Figura 38, pode ser acessada nas propriedades do *DataFlow Task* correspondente. Assim, estes componentes que precisam ser dinâmicos, como os componentes do tipo *Lookup* e *OLE DB Command*, devem estar devidamente adicionados na lista de expressões do respectivo *DataFlow Task*.

Nos itens desta subseção referentes às implementações realizadas nas etapas do processo ETL, são apresentadas as configurações dos componentes dos tipos origem e destino, que também precisam ser dinâmicos no modo de execução. Mas estes não são adicionados na lista de expressões, eles são configurados de forma dinâmica no próprio componente, usando os parâmetros do pacote.

Na Figura 39 é apresentada a implementação da tarefa do tipo *DataFlow Task* da entidade Cliente, desde o componente de origem, *OLE DB Source*, até o componente de destino, *OLE DB Destination*. Além disso, nesta figura, podem ser vistos dois agrupamentos que estão retraídos, o *CountNullFields Updated* e *Inserted*. Estes gru-

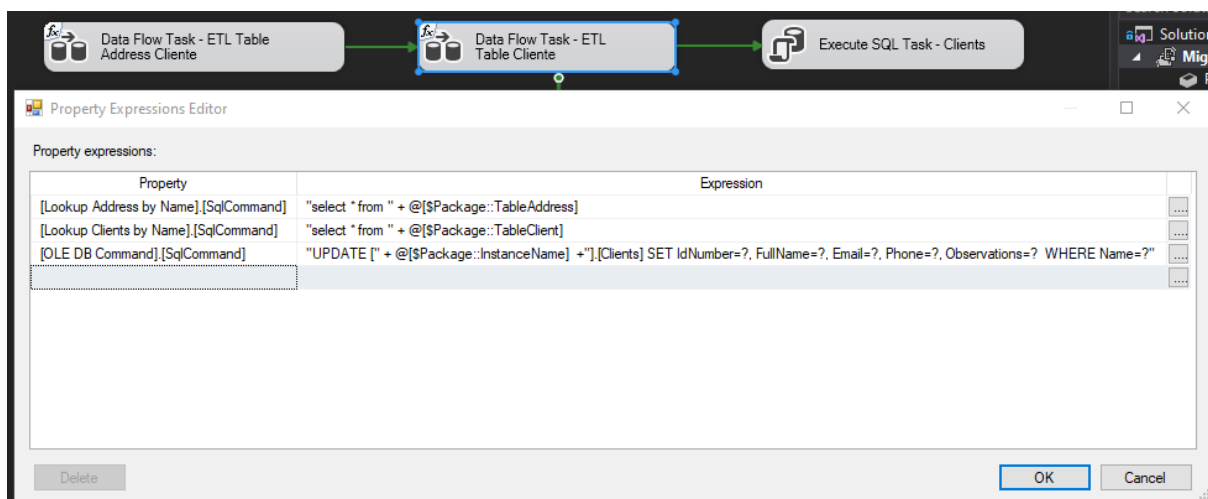


Figura 38 – Configuração das expressões do *DataFlow Task* da entidade Cliente
 Fonte: Elaborado pelo autor.

pos de componentes estão em fluxos diferentes, mas foram implementados de forma idêntica. A Figura 40 apresenta a implementação da contagem dos campos nulos no agrupamento do fluxo de novos registros.

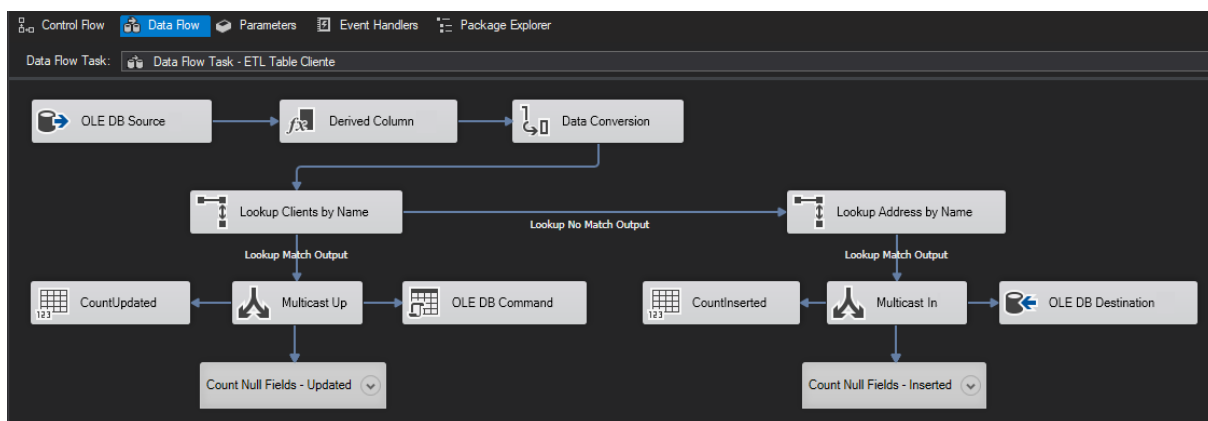


Figura 39 – *DataFlow Task* com os componentes da entidade Cliente.
 Fonte: Elaborado pelo autor.

Então, para implementar a contagem dos campos nulos foi preciso adicionar um componente do tipo Multicast, para poder distribuir os dados a vários componentes do tipo Split, que têm o propósito de filtrar os registros conforme condição adicionada em sua configuração, Figura 41. Neste caso, o fluxo *IdNumberNull* recebe os dados apenas se o campo *Copy of CPF_CNPJ* for nulo. Feito isso, a saída de cada componente Split é conectada a um componente do tipo *Row Count*, que, em sua configuração, deve ser associado a uma variável criada no pacote ETL para armazenar a contagem dos registros que passarem para o respectivo fluxo.

As variáveis configuradas nos componentes *Row Count* são enviadas para uma tabela na base intermediária (MAPDB), mas em outro esquema, *support*. Esta ação é executada após a finalização do respectivo *DataFlow Task* através de uma tarefa do

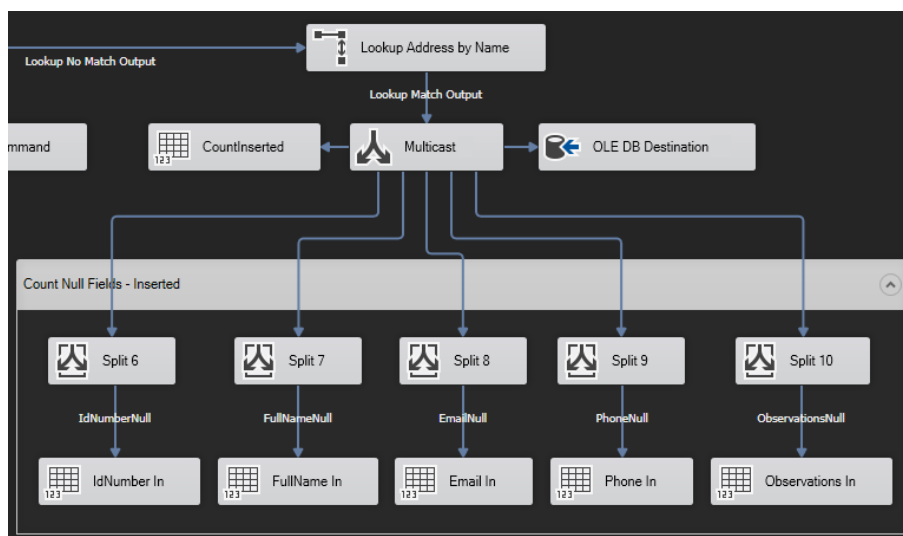


Figura 40 – Implementação da contagem dos campos nulos no pacote ETL
 Fonte: Elaborado pelo autor.

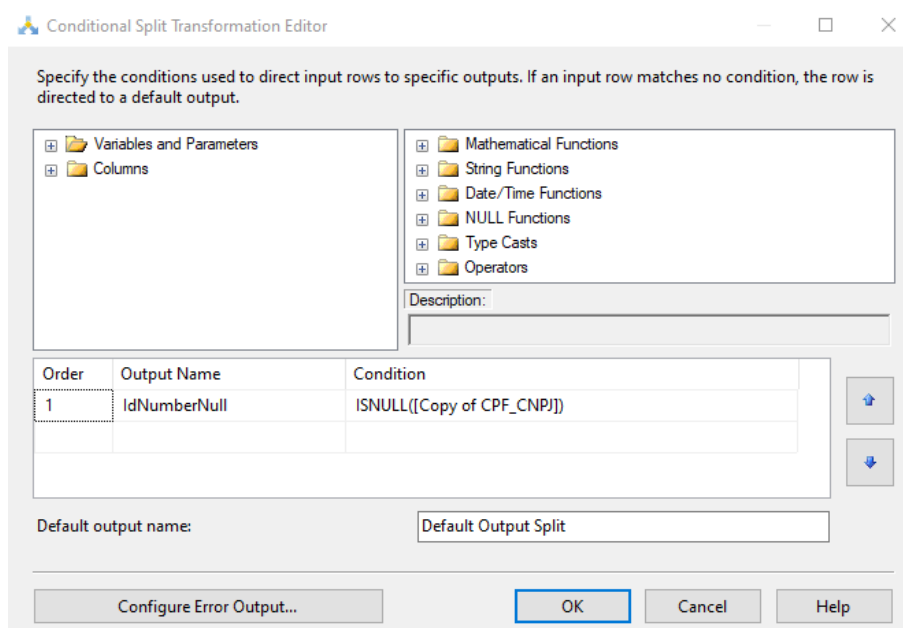


Figura 41 – Configuração do componente Split na contagem de um campo Nulo
 Fonte: Elaborado pelo autor.

tipo *Execute SQL Task*. A configuração desta tarefa pode ser verificada na Figura 42, que a princípio deve ser selecionada a base de dados e em seguida, o *script* SQL deve ser adicionado na propriedade *SQLStatement*, mostrado na Figura 43.

Após implementação dos processos de migração dos dados de todas as entidades, pode ser vista na Figura 44 a configuração final do *ControlFlow*. Da forma como foi configurado, cada entidade se encontra em um fluxo independente, desse modo, o processo ganha em agilidade, pois estes fluxos são executados de forma paralela.

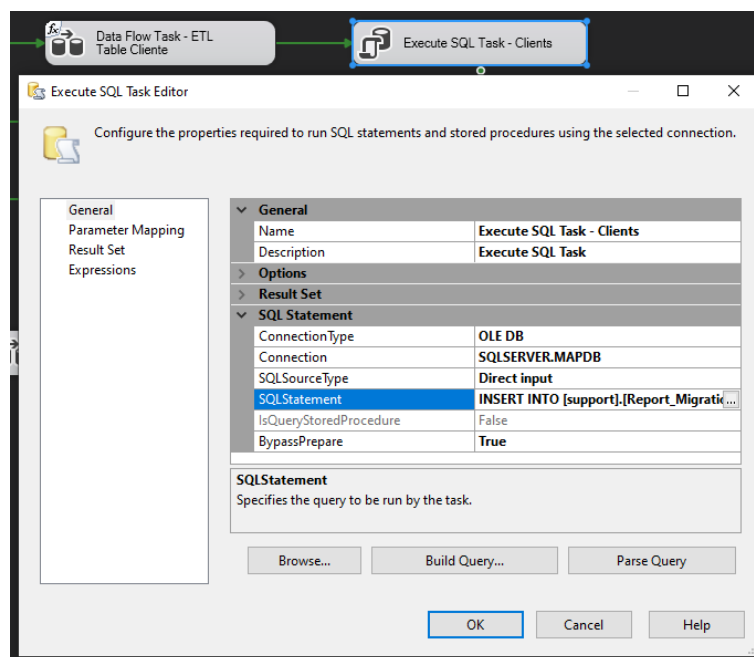


Figura 42 – Configuração da tarefa do tipo *Execute SQL Task*
 Fonte: Elaborado pelo autor.

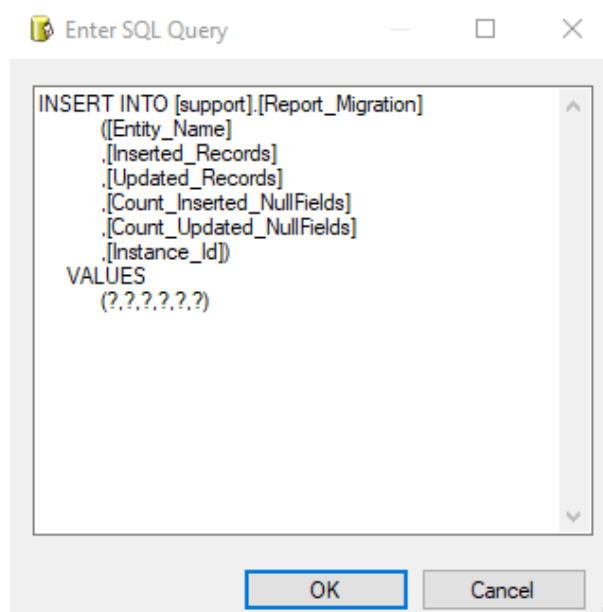


Figura 43 – *script* adicionado no *Execute SQL Task* para inserção dos valores das variáveis na contagem dos campos nulos. Fonte: Elaborado pelo autor.

Componentes de Origem

Neste trabalho apenas um tipo de componente de Origem foi usado, o *OLE DB Source*. Para configurá-lo, foi feita a escolha da base de dados, e em seguida, uma tabela desta base. Para a configuração da tabela foi necessário criar um parâmetro que é alterado no modo de execução. Como mostrado na Figura 45, *Instance_Id* foi o parâmetro adicionado, que corresponde ao identificador de cada empresa (instân-

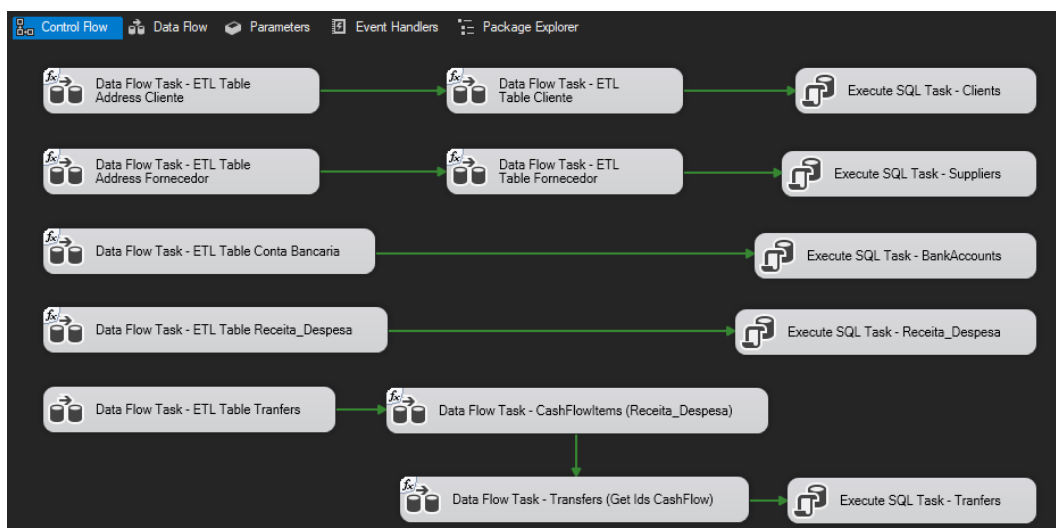


Figura 44 – ControlFlow da migração implementada no ambiente de desenvolvimento do SSIS. Fonte: Elaborado pelo autor.

cia) cadastrada no Flowup. Esse parâmetro foi criado para permitir que a ferramenta Migrup pudesse ser executada ao mesmo tempo por empresas diferentes, caso contrário, haveria conflito. Pelo lado da base MAPDB, todas as tabelas contêm um campo *Instance_Id* para carregar esse parâmetro no pacote ETL, e assim executar o *script (select)* apresentado na Figura 45 para carregar apenas os dados da empresa que está realizando a migração.

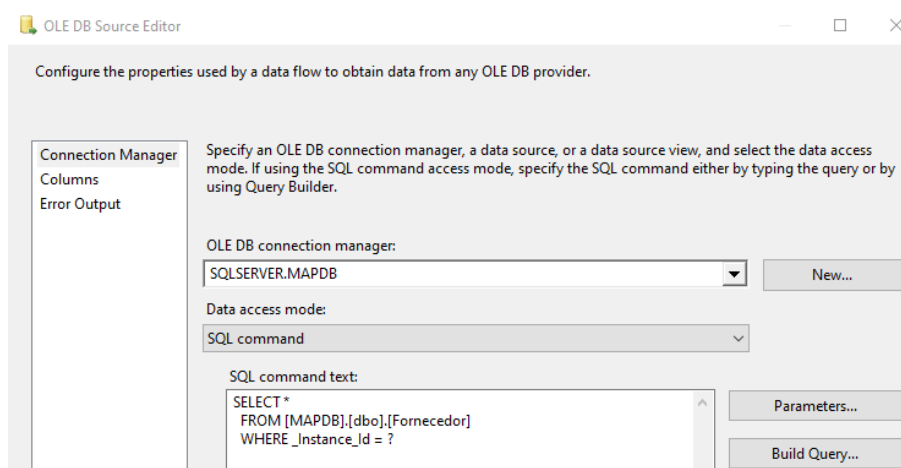


Figura 45 – Configuração da conexão com a base de origem, MAPDB. Fonte: Elaborado pelo autor.

Componentes de Transformação

Após a extração dos dados da base MAPDB, várias transformações foram necessárias para que estes dados fossem ajustados de acordo com as propriedades dos dados de destino na base do Flowup (Cash). Esta limpeza nos dados foi implementada nos componentes do tipo Transformação.

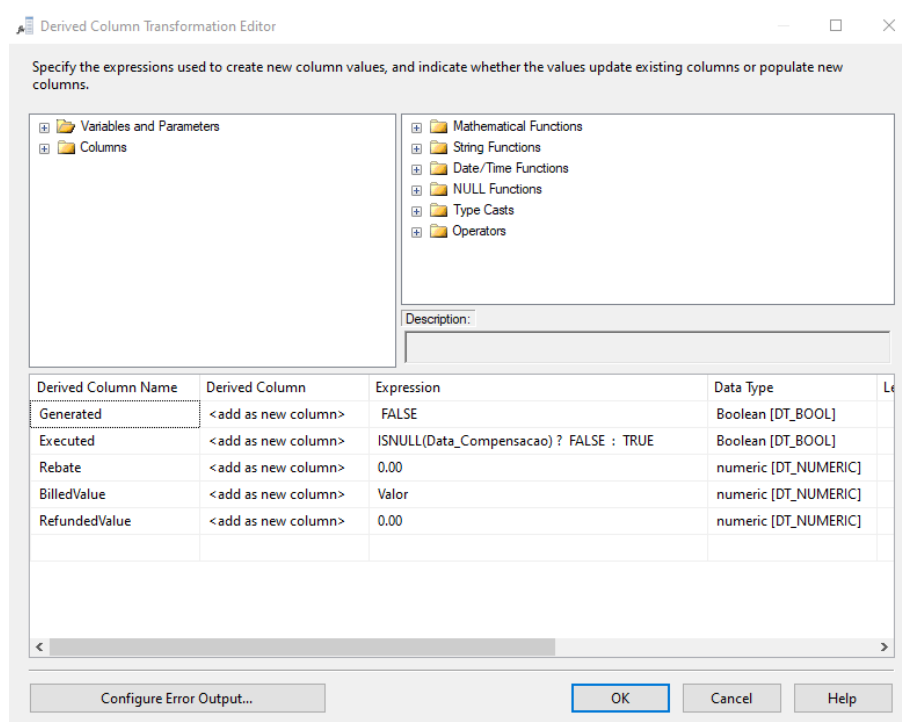


Figura 46 – Configuração do componente Derived Column
Fonte: Elaborado pelo autor.

Neste trabalho, os dois componentes geralmente usados logo após a extração foram: *DerivedColumn* e *Data Conversion*. A configuração do primeiro pode ser conferida na Figura 46, onde foram usadas expressões específicas das regras de negócio da tabela de destino. E na Figura 47 é mostrada a configuração do *Data Conversion*, com a conversão dos tipos dos dados conforme tipo da tabela de destino.

Na sequência vários componentes foram adicionados, sendo o *Lookup* um dos mais importantes. O primeiro passo na configuração deste componente neste trabalho foi adicionar uma expressão no *DataFlow Task* para dinamizar a seleção da tabela de destino que será consultada. Em seguida, como o exemplo mostrado na Figura 48, é feita a busca pelo nome do fornecedor, que é um campo único, para encontrar o identificador do Fornecedor na tabela de destino (*Suppliers*). Ou seja, se o valor consultado na tabela de destino configurada for encontrado, qualquer campo da tabela consultada pode ser capturado. Este campo, ou coluna capturada, é adicionada ao fluxo de dados (DataFlow), e poderá ser mapeada no componente de destino.

O *Lookup* também foi muito usado para verificar se o registro já existe na tabela de destino. Se já existe, os dados seguem para o fluxo de atualização, caso contrário seguem para o fluxo de novo registro. Para implementar dessa forma basta realizar a busca por algum campo único na tabela configurada como base de destino. Nestes casos não há necessidade de captura nenhum campo.

O componente *OLE DB Command*, que também é do tipo Transformação, foi

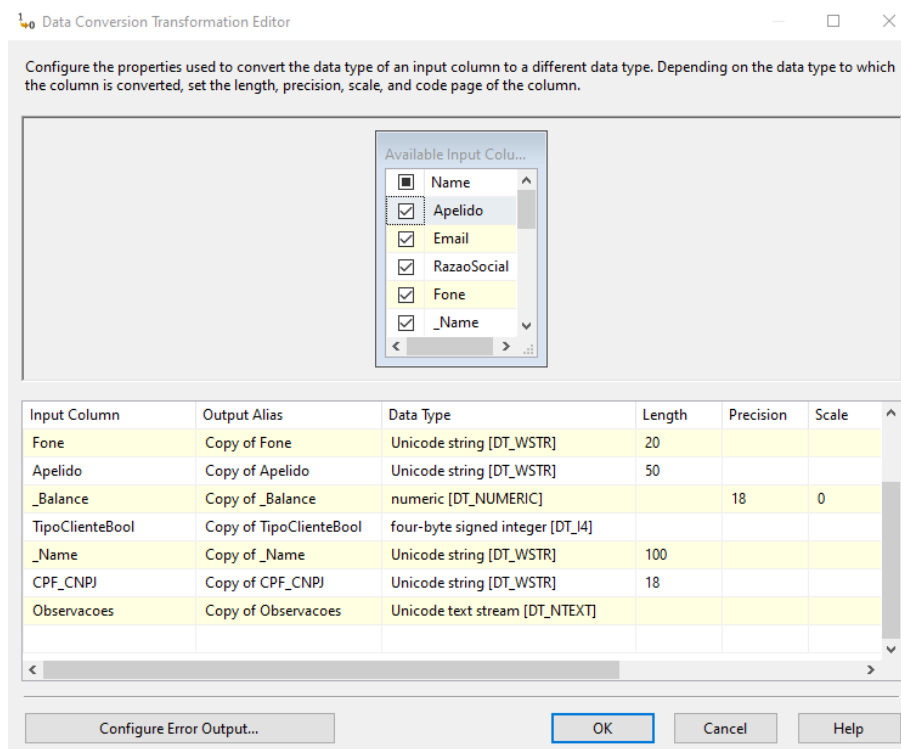


Figura 47 – Configuração do componente Data Conversion
 Fonte: Elaborado pelo autor.

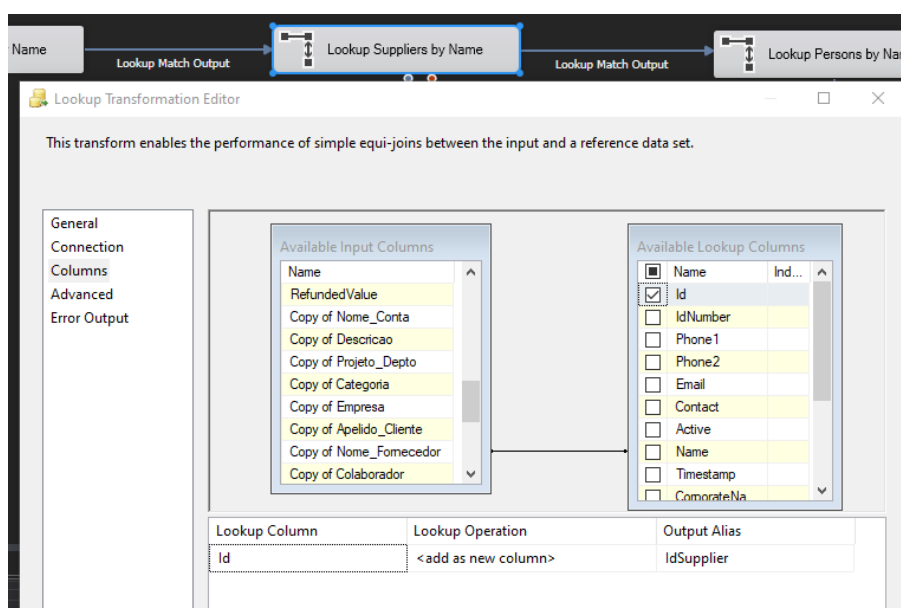


Figura 48 – Configuração do componente *Lookup* para pegar o identificador do fornecedor na tabela *Suppliers* no Cash. Fonte: Elaborado pelo autor.

utilizado no fluxo de atualizar os dados migrados, sua configuração é mostrada na Figura 49. São necessários três passos para configurar este componente: selecionar a base de dados, adicionar o *script* na propriedade *SqlCommand* e, por último, acessar a aba *Column Mappings* e associar os campos da tabela que está sendo usada no *script* com os parâmetros representados pelos símbolos de interrogação (?).

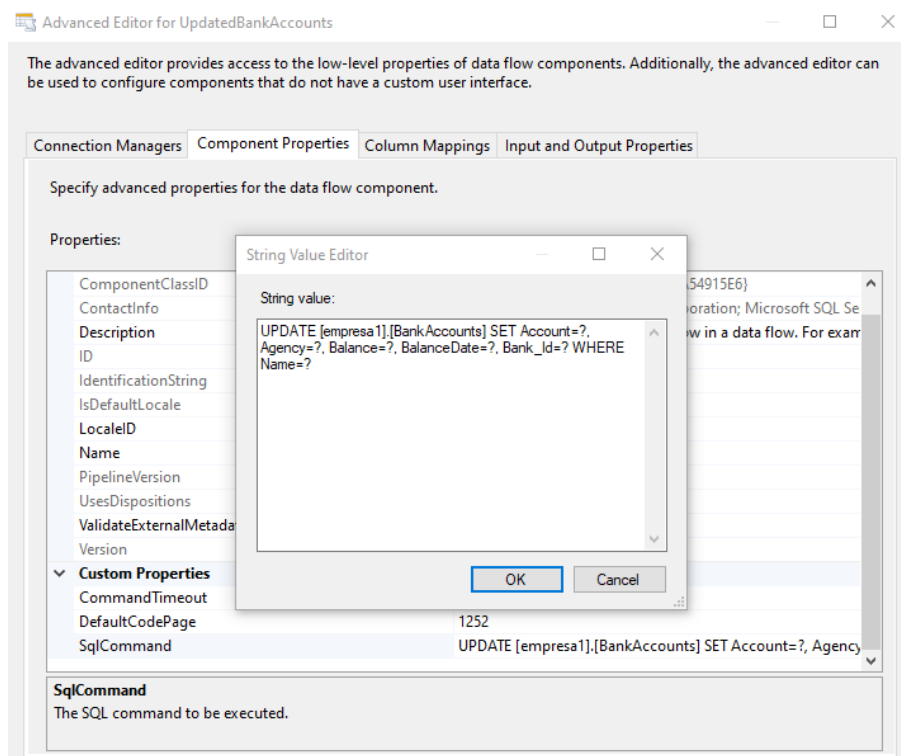


Figura 49 – Configuração do componente *OLE DB Command*
 Fonte: Elaborado pelo autor.

Componentes de Destino

O único componente de Destino usado foi o *OLE DB Destinations*, ele foi usado no final do fluxo do processo de migração de cada entidade para enviar os dados extraídos e transformados para a tabela de destino que corresponde à respectiva entidade. O primeiro passo na configuração destes componentes foi a seleção da base de dados, em seguida, para que fosse dinâmico, foi configurado um parâmetro na escolha da tabela, pode ser alterado no modo de execução. Como exemplo, na Figura 50 é mostrada a configuração deste componente na entidade Fornecedor.

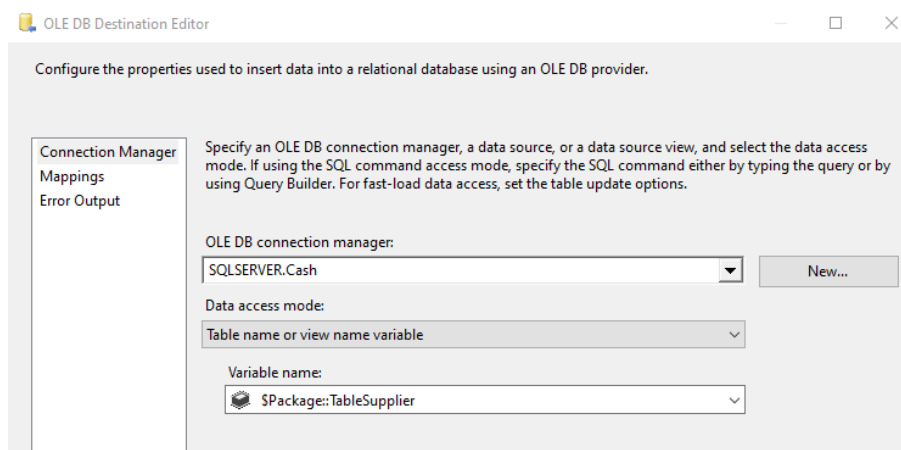


Figura 50 – Configuração da conexão com a base de destino, Cash
 Fonte: Elaborado pelo autor.

Após esta configuração inicial, foi realizado o mapeamento dos dados transformados com o dados da tabela de destino configurada. Na Figura 51, pode ser verificado o mapeamento no componente de destino da entidade Fornecedores.

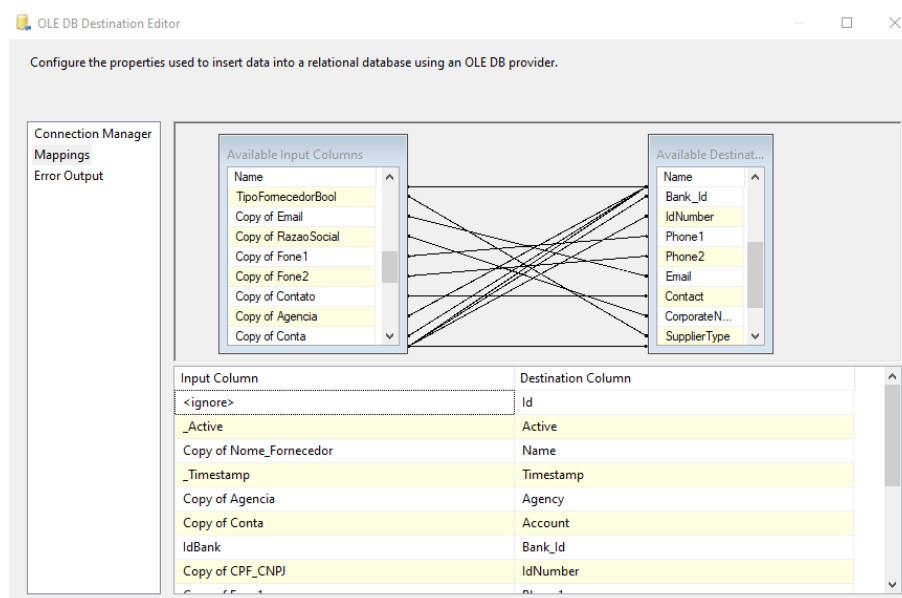


Figura 51 – Mapeamento dos campos da base de origem (MAPDB), já transformados, com a base de destino (Cash) da entidade Fornecedor. Fonte: Elaborado pelo autor.

A.2.4 Deploy do pacote ETL

Ao final da implementação, do pacote ETL, foi feita sua implantação (*Deploy*) no SQL Server, Figura 52. Nesta figura, à esquerda, podem ser verificadas as pastas e subpastas (*Integration Service Catalogs*, *SSISSB*, *Packages*, *Environments*, dentre outras), com também os próprios pacotes (*Mapping.dtsx* e *ImportBanks.dtsx*). E na janela maior, à direita, é mostrada a tela de configuração do pacote *Mapping.dtsx*, que disponibiliza a alteração das propriedades e na aba *Parameters*, os parâmetros podem ser associados às variáveis criadas na pasta *Environments*.

Dessa forma o pacote já implantado pode ser executado pela aplicação web, com o envio dos parâmetros.

A.2.5 Execução do Pacote ETL

Esta subseção apresenta os detalhes da implementação realizada neste trabalho para poder executar o pacote ETL a partir da aplicação web. Para tal fim, alguns passos foram necessários, como:

- Criação de parâmetros no pacote ETL;

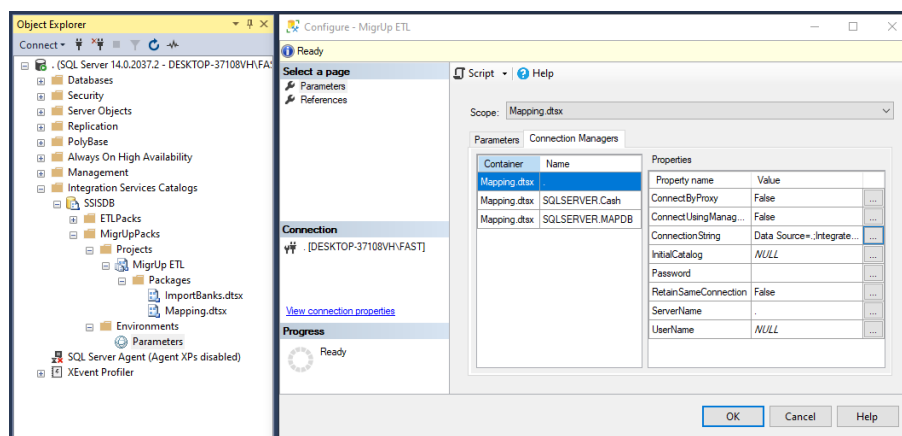


Figura 52 – Configuração do pacote ETL, disponível após o *Deploy*
Fonte: Elaborado pelo autor.

- Configuração dos componentes do *DataFlow Task* usando os valores dos parâmetros criados;
- Após o *Deploy*, foi criado um *environment*, nomeado de *Parameters*, na pasta *Environments* do Catálogo SSIS;
- Algumas variáveis foram criadas neste *environment Parameters*;
- Na configuração do pacote ETL, no Catálogo SSIS, estas variáveis criadas foram associadas ao seus respectivos parâmetros.

Após estes passos, o pacote ETL, implantado no Catálogo SSIS, ficou preparado para ser executado via aplicação web.

A.2.6 Relatório dos resultados

Os relatórios gráficos apresentados ao final do processo de migração dos dados, são carregados após consulta à tabela *[support].[Report_Migration]* na base intermediária (MAPDB). Esta tabela é carregada na etapa final de cada fluxo executado por uma entidade no *ControlFlow*, com os valores das contagens dos registros importados e a soma dos campos nulos de cada entidade.

A.3 Implementação das ferramentas auxiliares

Nesta seção são apresentados os detalhes da implementação das ferramentas auxiliares necessárias à ferramenta proposta.

A.3.1 Criação de uma base intermediária

Além dos motivos já citados nesta subseção, há outros relacionados ao painel de mapeamento, que contribuem na justificativa da criação da base MAPDB, como:

- Os nomes dos campos e tabelas da base do Flowup (Cash) estão em inglês; assim, estes nomes apareceriam no painel de mapeamento, porém, o sistema deve apresentar os textos na língua portuguesa;
- Alguns campos das tabelas da base Cash são chaves estrangeiras; isto poderia dificultar na associação entre os campos;
- Nem todos os campos de uma tabela na base Cash, são usados na migração; então, seriam desnecessários no painel de mapeamento;
- Foi preciso criar um campo extra em todas as tabelas da base MAPDB; isto não seria viável na base Cash;

Como exemplo demonstrativo, na Figura 53, são mostrados os campos das tabelas que representam a entidade *Receitas e Despesas*. Nesta figura, os campos à esquerda correspondem à tabela *[CashFlowItems]* na base de destino, Cash; e à direita, os campos da tabela *[Receita_Despesa]* na base intermediária, MAPDB.

CashFlowItems			
Column Name	Data Type	Nullable	
Id	int	No	
Date	datetime	Yes	
Description	nvarchar(200)	No	
Value	decimal(18, 2)	No	
Generated	bit	No	
Executed	bit	No	
TransactionId	nvarchar(MAX)	Yes	
Timestamp	datetime	No	
BankAccount_Id	int	No	
Category_Id	int	No	
Parent_Id	int	Yes	
Client_Id	int	Yes	
Company_Id	int	No	
CostCenter_Id	int	Yes	
Person_Id	int	Yes	
Supplier_Id	int	Yes	
Transfer_Id	int	Yes	
DueDate	datetime	Yes	
InvoiceNumber	nvarchar(200)	Yes	
CheckNum	nvarchar(MAX)	Yes	
RecurrentAsGenerated_Id	int	Yes	
ApprovedPayment	bit	Yes	
ApprovedPaymentDate	datetime	Yes	
ApprovedPaymentUser_Id	int	Yes	
BilledValue	decimal(18, 2)	No	

Receita_Despesa			
Column Name	Data Type	Nullable	
Id	int	No	
Data_Vencimento	datetime	No	
Descricao	varchar(200)	No	
Nome_Conta	varchar(100)	No	
Projeto_Depto	varchar(100)	Yes	
Categoria	varchar(100)	No	
Empresa	varchar(100)	No	
Apelido_Cliente	varchar(60)	Yes	
Nome_Fornecedor	varchar(100)	Yes	
Colaborador	varchar(100)	Yes	
Numero_NF	varchar(200)	Yes	
Data_Compensacao	datetime	Yes	
Data_Compentencia	datetime	Yes	
Valor	decimal(18, 2)	No	
_Timestamp	datetime	No	
_Instance_Id	varchar(10)	No	

Figura 53 – Comparação entre os campos de uma tabela da base de destino (à esquerda) e os campos de sua tabela correspondente na base intermediária (à direita).

Fonte: Elaborado pelo autor.

Importação da lista dos bancos

Para que os campos *Nome_Banco* e *Codigo_Banco*, presentes em duas tabelas do MAPDB, pudessem ser usados na importação, foi necessário atualizar a lista de bancos do sistema Flowup com os nomes e códigos dos bancos recentemente criados, como exemplo, Nubank.

Este subsistema usou a mesma estrutura já implementada para a solução proposta neste trabalho, porém, foi implementado em um outro ambiente do sistema Flowup, chamada de *ControlPanel*. Neste ambiente apenas a equipe do Flowup tem acesso. A ferramenta de importação legada também é executada no *ControlPanel*.

Nesta ferramenta de apoio foi usada a mesma abordagem da ferramenta geral, onde as fases de coleta dos metadados e mapeamento foram realizadas por codificação manual (*hand coding*). E para a fase de envio dos dados ao destino, foi utilizada a mesma ferramenta ETL, o SSIS.

Para a implementação das duas primeiras etapas, alguns passos foram realizados de forma bem semelhante ao processo geral:

- Criação de um novo esquema na base MAPDB, chamado de *support*, para separar os processos;
- Criação de uma tabela chamada *Banco* no esquema *support* (*[MAPDB].[support].[Banco]*), que possui os campos exigidos pela tabela da base de destino *Banks* (*[Cash].[global].[Banks]*); Nesse novo esquema há outras tabelas, uma para guardar os erros do processo e outra para os relatórios da migração;
- Adição do link no submenu do ambiente *ControlPanel*;
- Adaptação da interface de usuário;
- E foi feita também, a adaptação da implementação do mapeamento para importar corretamente a tabela baixada dos *dados abertos* (*Banco Central do Brasil*);

Na terceira etapa foi criado um pacote ETL específico para esta ferramenta de apoio. De forma análoga à ferramenta geral, a tabela *[MAPDB].[support].[Banco]* é populada após o mapeamento realizado. E na sequência, os dados desta tabela alimentam o pacote ETL criado para enviar estes dados à tabela de destino *[Cash].[global].[Banks]*.

Depois de finalizar a implementação do pacote, foi necessário fazer o *Deploy* para o SQL Server e chamá-lo da aplicação Web. A diferença é que neste caso não houve a necessidade de criar parâmetros de execução do pacote, pois não ocorre variação nos nomes das tabelas envolvidas, são fixos.

Então, para finalizar o processo desta ferramenta de apoio, foi necessário realizar alguns ajustes como: implementar o método na aplicação web para executar o pacote sem usar parâmetros; ajustar a interface gráfica de acordo com o padrão do *ControlPanel*, adicionando a indicação de carregamento (ícone *spin*) ao importar a tabela e ao executar a importação dos bancos; e adicionar o alerta com as mensagens ao final da migração.