INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-GRANDENSE - IFSUL, *CAMPUS* PASSO FUNDO
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

GUILHERME HENRIQUE PIASSON

ANÁLISE DE FERRAMENTAS DE INTEGRAÇÃO DE DADOS: UM ESTUDO DE CASO BASEADO NA FERRAMENTA PENTAHO DATA INTEGRATION

Evandro Miguel Kuszera

PASSO FUNDO, 2012

GUILHERME HENRIQUE PIASSON

ANÁLISE DE FERRAMENTAS DE INTEGRAÇÃO DE DADOS: UM ESTUDO DE CASO BASEADO NA FERRAMENTA PENTAHO DATA INTEGRATION

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-Rio-Grandense, *Campus* Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Me. Evandro Miguel Kuszera

PASSO FUNDO, 2012

GUILHERME HENRIQUE PIASSON

ANÁLISE DE FERRAMENTAS DE INTEGRAÇÃO DE DADOS: UM ESTUDO DE CASO BASEADO NA FERRAMENTA PENTAHO DATA INTEGRATION

Trabalho de Conclusão de Curso aprovado em ____/____/____ como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

_____

Orientador Prof. Me. Evandro Miguel Kuszera

_____

Prof. Me. Alexandre Tagliari Lazzaretti

_____

Profa. Esp. Carmen Vera Scorsatto

_____

Prof. Me. Evandro Miguel Kuszera

Coordenação do Curso

PASSO FUNDO, 2012

Aos meus pais, irmã, avós e amigos,
pelo apoio em minha
trajetória.

AGRADECIMENTOS

Agradeço aos meus pais Davi e Juassara, e irmã Vitória pelo apoio incondicional em minha trajetória de vida. As minhas duas avós por terem contribuído com os ensinamentos e terem se esforçado para me auxiliar nas dificuldades durante o período em que me mudei de cidade.

Juntamente com esses, agradeço a todos os meus professores ao longo de minha trajetória como estudante, por terem contribuído com o meu aprendizado teórico e com parte da construção do meu conhecimento de mundo.

Agradeço também, em especial ao meu professor orientador Evandro Miguel Kuszera, e aos demais professores do IFSul, por terem norteado os meus caminhos para a busca do conhecimento voltado para a área de tecnologia da informação, essa que até agora me fez seguir estudando e trabalhando de maneira cada vez mais motivadora.

“Quando se aprende as respostas, mudam-se as perguntas.”

Autor Desconhecido.

RESUMO

Este trabalho apresenta um estudo de caso implementado com o objetivo de efetuar um processo de migração de contextos de armazenamento. Esse estudo de caso baseou-se na migração de dados de um arquivo de controle de atividades de uma equipe pertencente a uma empresa desenvolvedora de software, para uma base de dados relacional. Para isso, foi efetuada uma análise a cerca de abordagens e metodologias de utilização de ferramentas ETL, com isso, foi realizada uma comparação entre as mesmas, onde foi escolhida a ferramenta *Pentaho Data Integration* para a construção do *workflow* de ETL.

Palavras-chave: integração de dados; ferramentas ETL; migração de esquemas de armazenamento.

ABSTRACT

This paper presents a case study implemented in order to perform a process migration context storage. This case study was based on the data migration of a file control activities of a team owned by a company developing software for a relational database. For this, an analysis was made about methodologies and approaches for the use of ETL tools, therefore, a comparison was made between them, which was chosen Pentaho Data Integration tool for building the ETL workflow.

Key words: data integration, ETL tools, migration of storage schemes.

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

BI – *Business Intelligence* – pág: 25.

CDC - *Change Data Capture* – pág: 24.

COBOL – *Commom Business Oriented Language* – pág: 29.

CRM - *Customer Relationship Management* – pág: 37.

E-LT –*Extract, Transform and Load*.– pág: 19.

EII - *Enterprise Information Integration* – pág: 19.

ERP – *Enterprise Resource Planing* – pág: 37.

ETL – *Extract, Transform and Load*– pág: 13.

GPL – *GNU Public License*– pág: 39.

JMS - *Java Message Service* – pág 37.

HTML - *HyperText Markup Language* - pág: 34.

KETTLE - *Kettle Extraction, Transport, Transformation and Loading Environment* - pág: 34.

KPI - *Key Performance Indicator* - pág: 35.

MER – Modelo Entidade Relacionamento- pág: 48.

IBM – *International Business Machines* - pág: 23.

ODBC - *Open Data Base Connectivity*- pág: 27.

OLAP - *Online Analytical Processing.* – pág: 34.

PDF –*Portable Document Format* - pág: 34.

PDI – *Pentaho Data Integration* - pág: 34.

T-EL- *Transform, Extract and Load* - pág: 19.

SGBD – Sistemas de Gerencia de Bancos de Dados – pág: 15.

WEKA – *Waikato Environment for Knowledge Analysis* - pág: 35.

SUMÁRIO

1 INTRODUÇÃO

A constante evolução nas tecnologias e estratégias utilizadas para o armazenamento de informações, paralelo ao aumento da utilização de recursos relacionados a tecnologia de informação, faz com que ao longo do tempo ocorra um aumento significativo no volume de dados gerados que necessitam de práticas que viabilizem a sincronização, o relacionamento, a integridade e a precisão nos dados armazenados.

Somado a isso, segundo a referência (DE SORDI & MARINHO, 2007) é perceptível o aumento de organizações dependentes do cruzamento de informações com vistas a aumentar a precisão no fornecimento de dados para auxiliar na tomada de decisões em tarefas dependentes de dados armazenados. Isto sendo baseado muitas vezes em conjuntos de dados armazenados com base em modelagens e mecanismos de gerência distintos.

Nesse estudo será realizada uma análise em torno da resolução de um problema real de uma empresa na área de desenvolvimento de software, a qual possui um esquema de controle de atividades que armazena dados na forma de planilhas. Logo, esse estudo visa realizar uma avaliação de ferramentas de integração de dados, que poderiam ser utilizadas no processo de migração do esquema antigo para um novo modelo descrito no estudo de caso. Somado a isso, abordar dificuldades, aspectos positivos ou negativos, funcionalidades, recursos e práticas a cerca de processos de integração de bases de dados. Isso com base no desenvolvimento de uma alternativa de migração do contexto antigo, para um contexto planejado que atenda a necessidades futuras da empresa.

1.1 MOTIVAÇÃO

O aumento significativo na utilização de recursos baseados em tecnologia da informação faz com que surjam novas necessidades quanto às formas de organização e armazenamento de massas de dados. Ao passo que a utilização de recursos informacionais ligados a um ambiente que necessita de armazenamento de dados aumenta, isso acarreta um aumento significativo do planejamento sobre a maneira de como esses dados devem estar dispostos, tanto na sua forma física (sistema distribuído ou centralizado), como de forma lógica, (referente aos mecanismos de armazenamento e gerência desses dados).

Nesse sentido, é possível que as informações acumuladas, sendo importantes, não devem ser descartadas, devem estar disponíveis e sincronizadas com novas informações que

surgem a cada instante. Isso deve ocorrer em determinados contextos, a exemplo de ambientes corporativos de organizações, que possuem diversas massas de dados, as quais são compostas por um ou vários esquemas, que podem ser formados por um ou vários bancos de dados. E estes, por sua vez, podem ter sido criados em momentos diferentes, com importâncias distintas, implicando em restrições variadas quanto aos dados armazenados, acarretando possíveis modelagens diferenciadas. Entretanto, todo esse conjunto pertencente a uma mesma organização pode necessitar de uma modernização, sem, em um primeiro momento, erradicar o contexto legado, ou até mesmo de um cruzamento de dados interno à organização para suprir uma determinada consulta. Somado a isso, talvez essa mesma organização possa vir a ser vendida ou se fundir com outra organização e essa estrutura complexa pode obrigatoriamente necessitar de um cruzamento de informações com um outro ambiente totalmente distinto quanto as suas regras de modelagem e gerência dos dados.

A partir disso, a motivação desse estudo consiste em uma análise exploratória na área de integração de bases de dados, procurando através de um estudo de caso, analisar problemas, pontos positivos e negativos a cerca de ferramentas de integração e abordagens, tendo como foco a utilização da ferramenta Pentaho Data Integration (CASTERS et. al., 2010). A qual foi escolhida para a implementação do estudo de caso, após a comparação entre as ferramentas.

Como contribuição, esse trabalho visa fornecer informações que podem ser utilizadas, para auxiliar na escolha de ferramentas de integração de dados, como também na elaboração de processos de integração com vistas a migração de esquemas de armazenamento.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Investigar o estado da arte dos mecanismos de integração de esquemas de armazenamento, como parte de um processo de migração de esquemas de dados, analisando ferramentas de ETL (*Extract Transform and Load)* existentes e escolhendo uma ferramenta para a aplicação e resolução de um estudo de caso real, envolvendo uma empresa de desenvolvimento de software.

1.2.2 Objetivos específicos

Investigar o estado da arte a cerca de mecanismos utilizados em processos de integração de esquemas de armazenamento.

Analisar ferramentas e metodologias utilizadas em integração de dados oriundos de sistemas de gerência e esquemas de armazenamento distintos.

Utilizar uma ferramenta via estudo de caso, com base na resolução de um problema real de uma empresa de software que armazena dados de uma maneira inapropriada. A qual possui um mecanismo de armazenamento que deve ser migrado para um novo modelo criado com base nas necessidades da organização.

Apresentar os resultados e contribuir com um estudo que pode servir de base para organizações que buscam alternativas para a migração de modelos de armazenamentos.

1.2.3 Organização do Documento

Este trabalho está organizado da seguinte forma: no Capítulo II será realizada uma introdução a abordagens e conceitos relacionados à integração e migração de contextos de armazenamento. No Capítulo III serão apresentadas as ferramentas de integração analisadas nesse estudo e seguidas de uma comparação, juntamente com a ferramenta escolhida para a implementação do estudo de caso.

No Capítulo IV será relatada a implementação do estudo de caso com base no contexto inicial de armazenamento, o processo de migração e por fim os resultados obtidos e uma discussão. Por fim, o Capítulo V apresenta o fechamento do trabalho, o qual é seguido das referências e dos anexos e apêndices.

2 FUNDAMENTAÇÃO TEÓRICA

Abaixo serão introduzidos os conceitos de dados e informação. Somado a isso, será feita uma análise a cerca de abordagens utilizadas perante a processos de integração e migração de contextos de armazenamento.

Segundo Date (DATE, 2004), o termo dado deriva da palavra *datu*, que significa "dar". Logo, dados ("fatos dados") podem ser utilizados para deduzir fatos adicionais. Informação, por sua vez, pode ser entendida como uma coleção de dados de forma ordenada, organizada e integrada de acordo com um determinado padrão.

Uma informação traz consigo um conceito, um valor agregado que serve para registrar um determinado fato. Portanto, quando se tem um conjunto relevante de informações também pode ser necessário um mecanismo que gerencie, organize, escalone e filtre essas informações de acordo com as exigências de quem as possui, levando em conta conceitos como credibilidade, confiabilidade, exatidão e segurança.

Por isso, para a resolução de determinados problemas, é que começaram a surgir os SGBDs (Sistemas de Gerência de Banco de Dados), os quais, em um primeiro momento apresentavam problemas quanto à ineficiência entre relacionamentos conceituais e o gerenciamento físico das informações, acabando por dificultar as formas de utilização desse recurso. A partir disso, surgiram os bancos de dados relacionais, os quais passaram a ser projetados com vistas a separar a representação conceitual, do armazenamento físico e também prover uma fundamentação matemática para esse contexto, somado a tecnologias de indexação, linguagens operacionais de alto nível e a utilização de ponteiros para o armazenamento físico, dentre outras particularidades que contribuíram para um melhor desempenho na manipulação de informações (ELMASRI, 2005).

No que tange à arquitetura física, desde meados dos anos 1960, quando em alguns casos os sistemas de banco de dados eram utilizados em plataforma mainframe, até os dias de hoje, ocorreram mudanças de paradigma. Pois antigamente os custos financeiros e a inconfiabilidade no hardware, composto por um conjunto distribuído de máquinas eram propícias à utilização de uma arquitetura centralizada, na qual massas de dados eram dispostas, muitas vezes, em um único recurso centralizado com um potencial de processamento satisfatório para a época. Somado a isso, também não era grande a quantidade de junção entre massas de dados oriundas de locais, e contextos distintos (ex.: cruzar dados entre bases de dados oriundas de mais de uma empresa). Com o passar do tempo, ocorreram melhorias quanto às possibilidades de hardware e acesso a tecnologias de armazenamento e

gerenciamento de informações, trazendo junto uma mudança em setores distintos da sociedade. A exemplo disso, muitos estabelecimentos comerciais passaram a implementar sistemas de gerenciamento de estoques, de vendas, de locação de produtos, entre outros, os quais acabaram por, aos poucos, substituir o armazenamento de determinadas informações no papel, para armazenar em meio eletrônicos. Isso caracterizando uma necessidade por recursos tecnológicos que fossem cada vez mais eficientes quanto ao armazenamento e gerenciamento de informações. Esse contexto, por sua vez, caracterizou as tecnologias de armazenamento e gerenciamento de informações como alternativa vital para empresas, universidades e entidades genéricas as quais passaram a utilizar esse tipo de recurso com vistas a modificar as formas rudimentares de detenção da informação para metodologias atuais, as quais trazem consigo organização, integridade, disponibilidade e segurança de seus dados.

A constante evolução dos recursos passíveis de utilização, oferecidos pela área e tecnologia da informação, proporcionou às organizações diversas soluções capazes de atender determinadas necessidades. No entanto, imersas em contexto dinâmico quanto a tecnologias, ferramentas, possibilidades e estratégias a serem tomadas as organizações podem adotar uma postura de vanguarda e atualizar suas práticas e metodologias quanto às formas de armazenamento através de projetos de migração de contextos.

Por outro lado, embora as técnicas de armazenamento e gerenciamento de dados foram evoluindo, muitas vezes, as mesmas podem ter que conviver com recursos antigos (sistemas legados), os quais em sua grande parte se fossem passíveis de extinção, demandariam de recursos financeiros, humanos, intelectuais e temporais para serem executados. Com isso, as organizações podem não acompanhar a evolução contínua do processo e permanecer com modelos utilizados, ou integrar contextos antigos com contextos atualizados, seja de maneira temporária durante o processo de migração, ou de maneira definitiva.

De acordo com o relatado anteriormente, é possível destacar algumas posturas a serem adotadas por organizações, no que diz respeito às decisões quanto ao armazenamento de dados (GARTNER, 2003).

2.1 Abordagem Tradicional e Conservadora

No que tange aos contextos de armazenamento que compõem uma organização, essa abordagem faz referência a uma integração duradoura entre partes heterogêneas, pois baseia-se na perpetuação de sistemas legados funcionando de maneira integrada a sistemas

modernos. Isso visando o funcionamento em conjunto, sem um prazo estabelecido de remodelação.

Como exemplo dessa abordagem, pode-se enquadrar organizações que possuem dados armazenados em planilhas do Excel, juntamente com dados armazenados em bases de dados Microsoft Access, Firebird, PostgreSQL Oracle e outros. Ambas provendo dados às necessidades da organização, cada uma com as suas particularidades e características, mas que estão ativas, sendo consultadas, alteradas e funcionando de maneira conjunta, para um ou mais sistemas.

Quanto a isso é evidente que em primeira instância um contexto como esse pode ser oportuno. Levando em consideração de que sistemas já existentes serão mantidos, evitando assim esforços referentes à melhoria, readequação e modificação na maneira como esses sistemas acessam e manipulam esses dados, provenientes de origem distintas, mas que já estão em funcionamento na organização.

Entretanto, com o passar do tempo, com o advento de novas tecnologias e o aumento no volume de dados, se mal planejada, essa postura tem grandes chances de ocasionar na geração de um macrosistema complexo, composto por partes heterogêneas com características distintas, acarretando no aumento de esforços em rotinas de manutenção ou até mesmo para a própria utilizaçãodos dados oriundos desse ambiente.

Perante esse cenário, pode-se citar alguns exemplos de problemas que essa abordagem pode trazer às organizações que a utilizam de maneira contínua:

- O surgimento de novas versões de ferramentas pode implicar no não fornecimento de suporte para versões antigas. O que desestimula investimentos e a continuidade em manter mão de obra especializada em uma tecnologia descontinuada.
- Diversidade de fabricantes de soluções construídas de forma genérica sem uma real adequação quanto às particularidades e as necessidade das organizações. O que pode ocasionar na construção e implantação de ferramentas que podem apresentar incompatibilidades ao serem implantadas de maneira individual, ou em conjunto com outras ferramentas.
- Dependência de mão-de-obra qualificada para uma determinada solução adotada em uma determinada época, que talvez em um determinado momento possa não fazer parte dos recursos disponíveis da empresa. E com isso, a organização pode não ter a quem recorrer, caso precisar.

Logo, estratégias que se enquadram nessa abordagem, podem ser mais indicadas para organizações que possuem sistemas com um nível alto de complexidade, ou que não dispõem de recursos para planejar e executar um projeto de migração. Outro exemplo de adoção dessa prática, pode ser em contextos temporários como parte de um projeto que visa uma abordagem sistêmica e evolutiva. Onde em um determinado momento é necessário realizar análises, transformações e validações a cerca de como os dados estão dispostos em todo o conjunto, para projetar um modelo composto por um depósito final que atenda as necessidades da organização. Após isso, com elaboração e finalização desse depósito, as partes heterogêneas podem ser desativadas e somente será utilizado o modelo final que foi construído.

Nessa ideia, podemos citar dois exemplos da utilização dessa abordagem de maneira temporária, um seria o estudo de caso deste trabalho, onde será efetuada a integração como etapa de migração de um contexto para outro, ou a construção de um *Data Warehouse* (IMMON, 2002*)*, baseado em dados oriundos de modelos de armazenamento distintos, que após a construção do depósito, poderão ser desativados.

2.2 Abordagem Sistêmica e Evolutiva

No que se refere à abordagem sistêmica e evolutiva, a mesma pode ser utilizada em processos de integração, com a ideia de modernizar e migrar um, ou modernizar, unificar e migrar vários modelos de armazenamento de uma organização.

Pois essa abordagem não prevê a postergação na utilização de modelos distintos de armazenamento, funcionando de maneira duradoura, deixando assim a organização suscetível aos problemas de um contexto como esse. Mas sim a readequação, transição e o posterior abandono de práticas antigas, com base em uma nova solução única de armazenamento.

Ao contrário da abordagem anterior, a abordagem sistêmica e evolutiva, pode demandar esforços quanto ao planejamento de práticas referentes as etapas realizadas ao longo de um projeto de migração de contexto.

Essas etapas podem consistir na elaboração de uma descrição a cerca das características e particularidades do(s) sistema(s) legado(s) (origem), a partir disso pode ser realizado um planejamento a cerca do objetivo final referente ao contexto de armazenamento (destino). Com isso em mente, a etapa de integração tem a obrigação de buscar os dados na origem, tratar e readequar esses dados e com isso submeter a carga no destino.

Por isso, a etapa de integração deve ser planejada com vistas a atender as necessidades esperadas a cerca do projeto final, e quanto a isso, essa etapa pode levar em consideração a utilização de ferramentas ETL (e respectivas variações de abordagem como ETL, E-LT, EII, entre outras), item que será abordado na seção seguinte.

2.3 Principais abordagens utilizadas em processos de integração

Ao longo do desenvolvimento desse estudo, por mais que a ênfase seja a integração momentânea entre origem e destino, como uma etapa no projeto de migração de contextos de armazenamento, é importante destacar os mecanismos e ferramentas que dão suporte a tarefas, no que se refere à integração e posterior migração de esquemas de armazenamento.

Quanto a isso, de acordo com as exigências particulares de cada projeto de integração, somadas à variedade de ferramentas disponíveis, é possível construir projetos de integração baseados não somente em uma ferramenta, associados a um *workflow* criado com etapas específicas.

Como exemplo disso, ao longo desse estudo serão analisadas mais de uma ferramenta de integração, associadas a descrições a cerca de formas de aplicabilidade das mesmas. Pois quanto a aplicabilidade, existem ferramentas que podem ser utilizadas em projetos que baseiam-se em *workflows* ETL e variações desse modelo (E-LT, T-EL, EII, ou outras sequencias de etapas criadas de acordo com as regras do projeto).

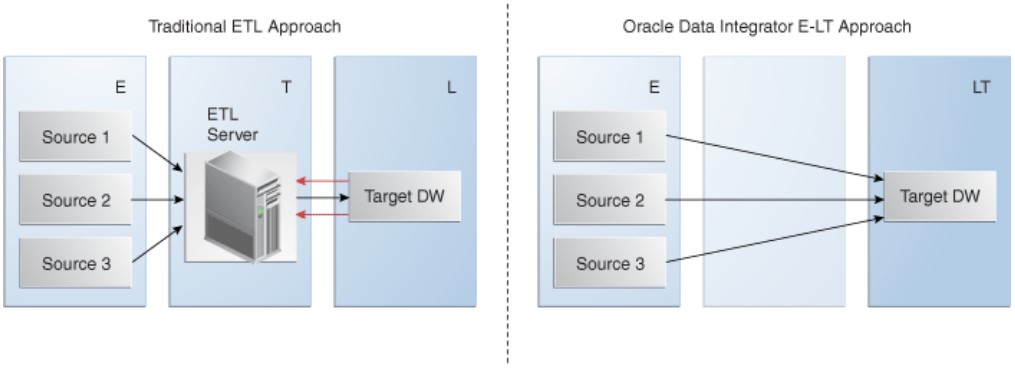
Ferramentas ETL: oriunda das palavras *Extract*, *Transform and Load*, essa terminologia diz respeito a ferramentas que viabilizam a extração de um conjunto de dados oriundos de uma ou inúmeras origens, transformação (readequação, validação do conjunto de dados extraídos, exclusão e ou geração de novos valores) e persistência de informações de um local para outro. Ou seja, esse tipo de ferramenta pode ser utilizado em tarefas de modernização, readequação, validação e integração de massas de (KIMBALL & CASERTA, 2004).

No que se refere a ferramentas de integração, além das ferramentas de ETL, conforme citado anteriormente, também existem projetos que usam variações desse conceito, dentre os quais são citadas algumas abaixo:

- E-LT (*Extract, Load and Transform*): Processos que utilizam essa abordagem eliminam a necessidade de uma etapa intermediária, de um servidor independente com a finalidade de realizar transformações, validações e readequações nos dados entre a origem e o destino. Com isso, essas tarefas são

realizadas no destino de uma forma mais próxima ao contexto e ao SGBD do depósito final de dados, podendo realizar a extração na origem, transferir os dados para o destino e por fim realizar o tratamento e realocação dos dados dentro do depósito de destino dos dados, a exemplo da Figura 1.

Figura 1- Comparação do uso de ferramentas ETL e ferramentas E-LT.



Fonte: http://download.oracle.com/docs/cd/E14571_01/integrate.1111/e12643/img/elt_approach.gif

- T-EL (*Transform, Extract and Load*): essa topologia é diferenciada dos modelos de ETL e E-LT, devido ao fato de que a etapa de transformação e validação dos dados pode ser realizada na origem. Após isso, realizando as etapas de extração dos dados já adequados, padronizados e validados da origem e posterior carga no destino. Prática essa que pode facilitar as tarefas de extração e carregamento dos dados no destino, considerando que os dados extraídos da origem devem estar em sua maioria prontos para a carga. Dessa forma, exigirão menos processamento em etapas de transformação fora da origem. Entretanto essa prática pode não ser adequada para contextos, em que as transformações na origem torne-se algo complexo, e utilize processamento além do permitido.
- EII (*Enterprise Information Integration*): esse modelo faz referência à práticas diferentes das adotadas no escopo desse estudo, as quais baseiam-se em processos de integração e modificação física do esquema de armazenamento dos dados. Pois em contextos que se baseiam na topologia EII é importante ressaltar que os dados tornam-se acessíveis através de processos de integração, somente no instante em que determinados requisições são solicitadas.

  Pois nesse estudo será abordada uma prática definitiva quanto aos valores a localização física dos dados. Entretanto, nas abordagens EII, é criado um

contexto virtual e momentâneo de integração dos dados (CASTERS et. al., 2010).

No que tange a utilização dessas ferramentas em processos de integração, de acordo com (FERREIRA et. al., 2010), foi realizado um estudo e com isso, gerada uma tabela a cerca da utilização de ferramentas ETL e uma linha do tempo:

Tabela 1 – Utilização de ferramentas ETL em uma linha do tempo

Ano	***Título***	***Significado***
Início de 1990	Codificação manual de ETL	Códigos personalizados escritos à mão
1993-1997	A primeira geração de ferramentas de ETL	Código baseado em ferramentas de ETL
1999-2001	Segunda geração de ferramentas de ETL	Código baseado em ferramentas de ETL
2003-2010	Ferramentas de ETL atualmente	A maioria das ferramentas eficientes

Fonte: Conforme o trabalho de (FERREIRA et. al., 2010).

Nessa ideia, as organizações podem construir uma solução própria desenvolvida em uma determinada linguagem de programação e que atenda as suas necessidades. Por outro lado, podem fazer uso de ferramentas já construídas, utilizadas e consolidadas no mercado, as quais dentre suas funcionalidades, apresentam soluções para tarefas de integração.

3 FERRAMENTAS DE INTEGRAÇÃO

3.1 Oracle GoldenGate

O Oracle GoldenGate surgiu após a aquisição pela Oracle da empresa GoldenGate Software em 2009, de acordo com notícia vinculada ao site da Oracle (ORACLE, 2012) .

Atualmente pertence à suite de aplicativos *Fusion Middleware* da Oracle, suíte esta que fornece um conjunto amplo e variado de ferramentas que podem atender às necessidades de corporações que necessitem de soluções referentes a controle de processos, gerenciamento de dados, integração de ambientes, entre outros.

De acordo com John (JOHN, 2011), essa ferramenta visa a construção de projetos baseados na integração (captura, filtragem, transformação, replicação de dados e transações) entre ambientes heterogêneos de maneira contínua e em tempo real.

A mesma é constituída por componentes que viabilizam determinadas funcionalidades, de acordo com a tabela abaixo:

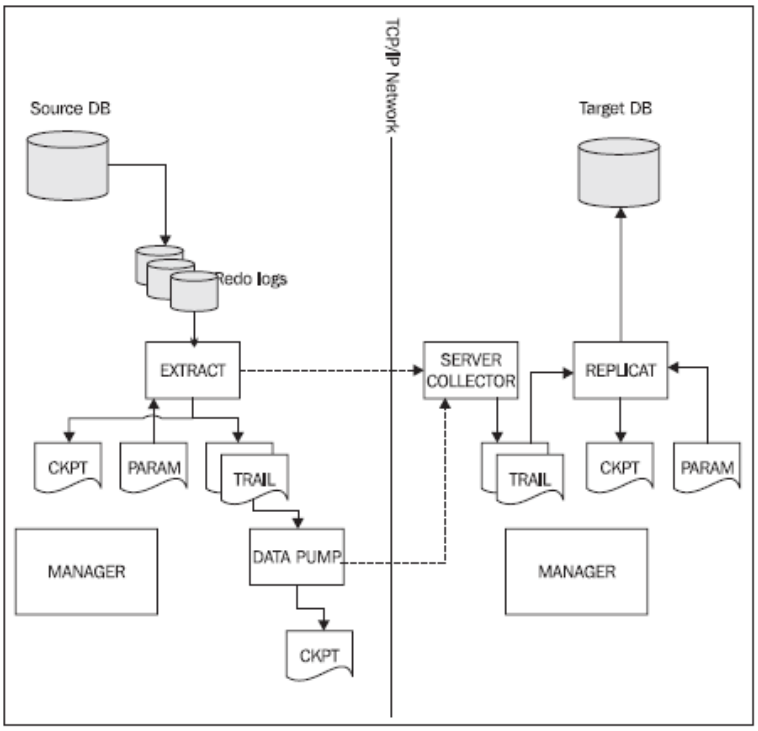
Tabela 2– Componentes da ferramenta GoldenGate

Recurso	***Descrição/Funcionalidade***
CheckPoints	Utilizado na marcação de status dos processos.
DataPump	Utilizado para o envio de informações entre partes do sistema.
Extract	Compõe o mecanismo de captura de informações.
Manager	Utilizado no controle de processos.
Replicat	Utilizado na leitura (trails) e encaminhamento dos dados aos sistemas destino.
Server Collector	Recebe os dados do Extract e Data Pump, escreve alterações nos trails.
Trails	Arquivos que armazenam operações realizadas no processo.

Fonte: Conforme John (JOHN, 2011).

Abaixo segue a Figura 2, com a finalidade de exemplificar os componentes e as suas finalidades dentro de um processo de integração.

Figura 2 - Relação entre componentes da ferramenta GoldenGate



Fonte: Conforme John (JOHN, 2011) – pág 16.

No que tange a implantação da ferramenta, a mesma possui uma arquitetura flexível e modular, que possibilita a escalabilidade e adaptabilidade de acordo com o desenvolvimento e execução do projeto.

Somado a isso, com relação às topologias de implementação, a mesma pode ser utilizada para replicação de bases dados, balanceamento de carga, integração unidirecional (um ponto a outro), integração convergente (várias origens e um destino), entre outras.

Apresenta uma ampla variedade quanto as possibilidades de contextos de entrada e saída dos dados (IBM DB2, Oracle, MySQL, Microsoft SQL Server, Sybase, Teradata, *web services*, entre outros) e plataformas de hardware (Linux, Windows, Solaris, HP-UX, HP Tru64, IBM AIX, IBM Z/OS, entre outros), de acordo com a matriz de drivers de conexão do GoldenGate (ORACLE, 2012).

Com relação ao carregamento dos dados no(s) destino(s), é possível utilizar técnicas de paralelismo, agrupamento de pequenas transações, entrega em lotes, entre outras.

Outro ponto importante é que essa ferramenta possibilita a geração de dados transacionais dos processos em tempo real, para fins de acompanhamento. Além disso, é dotada de mecanismos de recuperação de processo em caso de problemas de indisponibilidade ao longo das etapas de integração.

3.2 Oracle Data Integrator

Concebida pela Oracle no ano de 2006, após a aquisição da empresa francesa Sunopsis, de acordo com notícia vinculada ao site da Oracle (ORACLE, 2012), e atualmente também pertence ao conjunto de soluções de *Fusion Middleware* da Oracle.

De acordo com as duas documentações (*Oracle Fusion Middleware Getting Started with Oracle Data Integrator* e *Developer's Guide for Oracle Data Integrator*) de Miquel (MIQUEL, 2011), pode-se dizer que o ODI é uma ferramenta que fornece suporte à integração de dados através de uma solução unificada com vistas ao planejamento, construção, implantação e gerenciamento de um depósito final de dados.

Ao construir um *workflow* de ETL nessa ferramenta, devem ser levados em consideração a possibilidade de integração baseada em três estilos: integração orientada a eventos, integração orientada a serviços e integração orientada a dados (em lote).

As tarefas que compõem um *workflow* podem ser baseadas em módulos de conhecimento (*knowledge modules*), dentre os quais podemos citar:

- CKM (*Check Knowledge Module*) - utilizado em tarefas de verificação e validação ao longo do *workflow.*
- IKM (*Integration Knowledge Module*) – utilizado na integração dos dados para serem carregados no destino.
- JKM (*Journalizing Knowledge Module*) – utilizado na criação de estruturas (ex.: modelo de dados, CDC (*Change Data Capture*), entre outros).
- LKM (*Load Knowledge Module*) – utilizado durante a carga de dados em *data servers* distintos.
- RKM (*Reverse Knowledge Module*) – utilizado em tarefas de engenharia reversa de modelos de dados (ex.: recuperação de metadados).
- SKM (*Service Knowledge Modules*) – utilizado para manipulação de dados via *web services*.

Outro componente importante dentro da arquitetura do ODI são os repositórios, que podem consistir em repositórios mestres e repositórios de trabalho. Os repositórios mestres têm a finalidade de armazenar os usuários com suas respectivas permissões de acesso à plataforma, definições de servidor, esquemas, versões de processos desenvolvidos, entre outros. Por outro lado, os repositórios de trabalho têm a função de armazenar regras de negócio, *logs*, blocos de dados, definições de colunas, pacotes, *procedures*, entre outros.

Além dos repositórios, outros componentes importantes são os agentes e os cenários, os agentes (chamados de *Run-Time Agent)* podem ser utilizados em tempo de execução do processo de integração através da criação de cenários a partir de determinados processos os

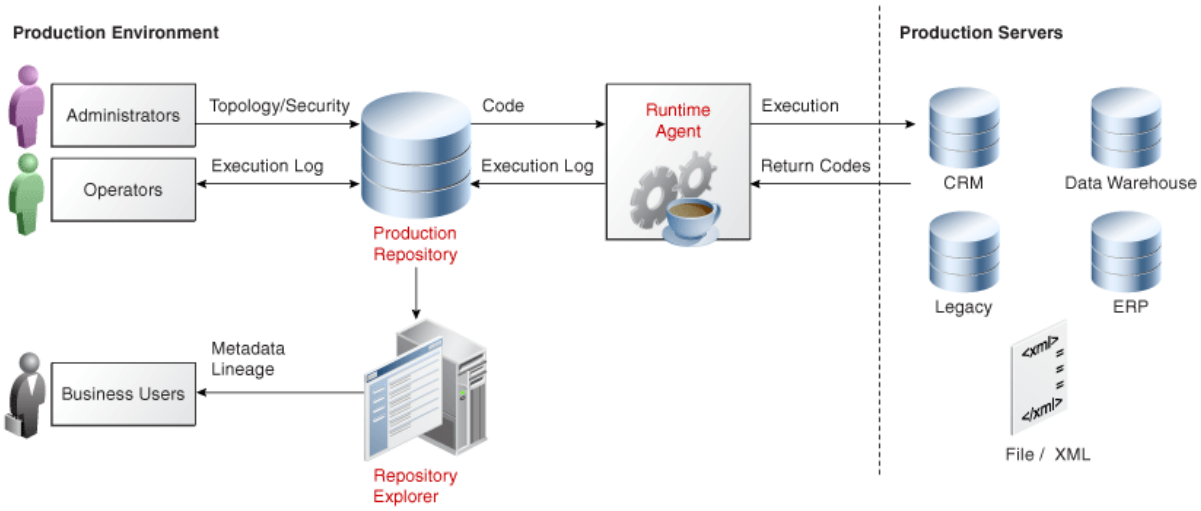
quais podem gerar códigos de retorno (ex.: número de registros processados, tempo de execução da tarefa), mensagens, *logs*, entre outros, os quais por sua vez, ficam armazenados em um determinado local e podem ser recuperados pelo *Run-Time Agent*.

Quanto as conexões, essa ferramenta comunica-se com soluções de armazenamento como: Oracle, Teradata, IBM DB2, Informix, IMS DB, VSAM Batch, Adabas, Microsoft SQL Server, Netezza, Sybase, Ingres, MySQL,Postgres, Interbase, entre outros, conforme a matriz de drivers de conexão do ODI (ORACLE, 2012).

No que se refere à qualidade dos dados transferidos ao longo do processo de integração, o ODI pode contar com a utilização de ferramentas como o *Oracle Data Quality* e *Oracle Data Profiling Integration*, as quais possibilitam a análise e o monitoramento da qualidade dos dados baseados em medidas coletadas ao longo do processo de integração.

Abaixo, (conforme a Figura 3), segue um exemplo da arquitetura funcional da ferramenta:

Figura 3 - Arquiteura funcional da ferramenta



Fonte: Baseado em exemplo existente em http://docs.oracle.com/cd/E21764_01/integrate.1111/e12643/intro.htm

3.3 IBM Cognos (Data Manager)

O IBM Cognos é uma ferramenta concebida pela IBM após a aquisição da empresa Cognos em 2007, conforme notícia vinculada ao site da IBM (IBM, 2012).

Nesse contexto, o plantel de ferramentas oferecidas para projetos de BI (*Business Intelligence*) sofreu incorporações de softwares então adquiridos após o processo da empresa Cognos. Dessa forma, a ferramenta IBM Cognos é enquadrada como uma ferramenta de BI,

que tem a sua composição baseada em mais de uma ferramenta, dentre as quais, esse estudo dará ênfase à ferramenta IBM Cognos Data Manager (antiga Cognos DecisionStream).

No que se refere a essa ferramenta, a mesma provê funcionalidades de ETL (pois é utilizada em etapas extração, transformação e carga dos dados), composta pelos seguintes componentes, de acordo com o centro de informações do *IBM Cognos Business Intelligence* (IBM, 2012):

- *Data Manager Designer:* componente o qual é utilizado como uma interface para a construções de processos que acessam dados, os transformam e carregam-nos em um destino (ETL). As informações desenvolvidas nesse componente são armazenadas em um *Cognos Data Manager Catalog*. Obs.: O componente *Data Manager Designer* pode ser instalado somente em plataforma Windows.
- *Data Manager Configuration*: é um componente utilizado para a configuração de recursos, iniciar e parar serviços. Um exemplo de sua aplicabilidade é a configuração do IBM Cognos Data Manager para que funcione com base em componentes como: *Cognos Data Manager Network Services, IBM Movement Sevice* e outros componentes da suíte de BI.
- *Data Manager Engine*: esse componente consiste em um conjunto de programas que podem ser executados através do *Data Manager Designer* ou via linha de comando.
- *Data Manager Network Services*: utilizado para a execução de fluxos, workflows ou JobStream em locais remotos. Esse componente inclui um servidor instalado juntamente com o IBM Cognos Data Manager Engine e também provê mecanismos de auditoria, quando são executadas determinadas tarefas. Um exemplo de aplicabilidade desse componente seria em contextos em que o usuário acessa o *Cognos Data Manager Designer* em um host diferente do que está instalado o *Cognos Data Manager Engine.*
- *Data Movement Service*: permite aos usuários a execução e agendamento de execução de *workflows* em locais remotos usando o recurso IBM Cognos Connection (interface para a suíte IBM Cognos *Business Intelligence*).

  Para esse componente funcionar, o *IBM Cognos Data Movement* deve ser instalado com o *IBM Cognos Data Manager Engine*, sendo esse último instalado no mesmo host que se localiza o IBM Cognos BI Server.

- *Data Manager Connector for SAP R/3:* esse componente é utilizado para viabilizar a interação dessa ferramenta com fontes de dados SAP R/3.

Como características da ferramenta, podem ser destacados os seguintes tópicos:

- No que se refere à topologia os componentes da ferramenta podem ser instalados de variadas formas (um ou vários servidores).
- Pode ser utilizado de forma associada a mecanismos como *web services* e outros para otimizar os processos de integração e receber ou enviar dados para um meio externo do *workflow*.
- Armazena informações a cerca de processos de extração, transformação e carga em um catálogo.
- Possui mecanismos de testes e verificações de etapas individuais do *workflow*.
- Pode ser utilizada como um componente que faz parte a suíte de BI IBM Cognos, o que significa que a mesma pode funcionar de maneira associada com outras ferramentas como: *IBM Rational Insight* (o qual pode ser utilizado para a geração de relatórios), entre outras.
- É composto por drivers de conexão, de acordo com os drivers de conexão em IBM (IBM, 2012), que possibilitam a extração e carga de dados oriundos de ambientes heterogêneos como: DB2, DTS Package, Essbase, INFORMIX, ODBC, Oracle, Published FM Package, Microsoft SQL Server, SAP R/3, SQLTXT, Sybase, TM1, entre outros.
- Possui mecanismos de acompanhamento e análise do desenvolvimento das etapas de integração.
- Possui arquitetura escalável de acordo com o volume de dados envolvidos, demanda de processamento e outras necessidades ao longo da execução do processo.

3.4 Informatica PowerCenter

Pertencente à empresa Informatica, a ferramenta PowerCenter é um dos principais produtos dessa companhia que foi fundada em 1993 e que é conhecida no mercado de ferramentas ETL como uma das ferramentas mais consolidadas e que tem o seu foco principal na atuação em projetos de ETL e não somente BI.

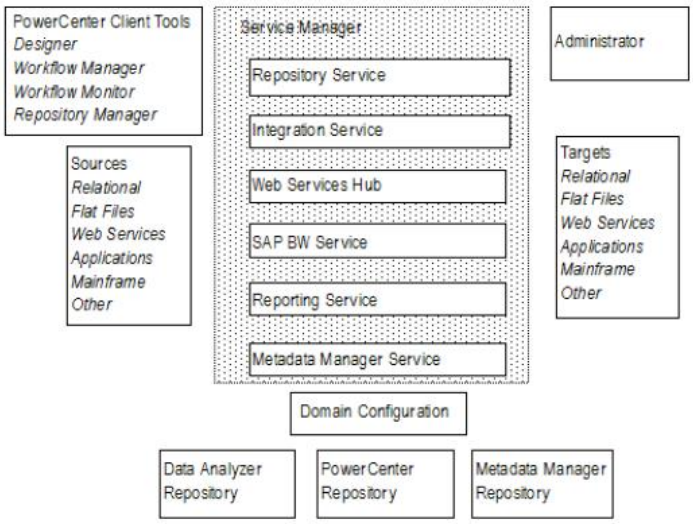
A ferramenta é formada principalmente pelos seguintes componentes (INFORMATICA, 2012):

- *Designer:* utilizado para a criação e manipulação de maneira gráfica as instruções e etapas ao longo de um *workflow*.
- *Workflow Manager:* sua principal finalidade é possibilitar a criação, agendamento e execução de *workflows.* Os quais são compostos por tarefas relacionadas à extração, manipulação e carga de dados.
- *Workflow Monitor:* utilizado para visualizar, monitorar o agendamento e o andamento da execução de *workflows.*
- *Repository Manager*: componente utilizado para o gerenciamento do repositório. Com isso, podendo manipular projetos, gerenciar usuários e grupos, entre outros.
- *Domain*: é a principal unidade para o gerenciamento e administração de *workflows* criados no PowerCenter. Dentro do *domain* existem *nodes* (uma representação lógica), e dentro de um node existe o *Service Manager.*
- *Data Analyzer Repository*: a principal finalidade é a de armazenar informações sobre objetos e tarefas, para a posterior geração de relatórios.
- *PowerCenter Repository*: consiste em um banco de dados utilizado para o armazenamento de informações criadas ao longo do processo e que são necessários para a execução de instruções.
- *Metadata Manager Repository*: utilizado para armazenar informações a cerca de metadados.
- *Administrator*: também chamado de *Administration Console*, esse componente provê uma interface web para a administração de recursos.
- *Repository Service:* busca informações no *PowerCenter Repository* e envia para outros componentes.
- *Integration Service:* sua finalidade principal é extrair dados, processá-los e remetê-los para os alvos.
- *Web Services Hub:* utilizado para a comunicação do *workflow* construído no Power Center, com *web services* externos.
- *SAP BW Service*: atua na extração e carregamento de dados com o SAP NetWeaver BI.
- *Reporting Service:* utilizado para processos de análise.

- *Metadata Manager Service:* sua finalidade é a de executar a aplicação *Metadata Manager*.

Quanto a arquitetura da ferramenta, a Figura 4 tem a finalidade de exemplificar a organização de seus componentes.

Figura 4 - Componentes da ferramenta PowerCenter



Fonte: (INFORMATICA, 2012) – pág 2.

No que se refere à extração de dados, essa ferramenta se comunica com SGBDs como: Oracle, Sybase ASE, Informix, IBM DB2, Microsoft SQL Server, and Teradata. Somado a isso, podem ser lidos arquivos em COBOL, XML, Microsoft Access, Microsoft Excel, *web services* e acrescentadas as conexões do *Informatica PowerExchange*.

Por outro lado, a ferramenta não permite o carregamento de dados em arquivo COBOL, e em planilhas do Excel. No entanto permite o destino de dados através de sistemas que usam drivers ODBC e serviços FTP.

Quanto a características, a mesma possibilita uma ampla gama de recursos quanto a projetos de integração no que se refere à escalabilidade conforme a demanda, segurança e mecanismos de recuperação de falhas, sistemas distribuídos e afins. Somado a isso, também provê mecanismos de tratamento dos dados, monitoramento de processos e geração de relatórios.

3.5 SAP BusinessObjects Data Integrator

Após a aquisição da empresa francesa Business Objects pela SAP em 2007, conforme notícia de aquisição (SAP, 2012), a suíte de funcionalidades BusinessObjects passou a fazer parte dos produtos da suíte de BI da SAP (*Business Objects Business*

Intelligence). Nesse estudo, será levado em conta uma de suas ferramentasa qual é chamada de *SAP BusinessObjects Data Integrator*.

A ferramenta *BusinessObjects Data Integrator* é uma ferramenta utilizada para tarefas de integração e movimentação de dados, podendo ser enquadrada como uma ferramenta para processos de ETL. Dessa forma, a mesma é constituída pelos seguintes componentes, conforme a documentação *Data Integrator Getting Started Guide* (SAP, 2012):

- *Data Integrator Access Server*: utilizado para a coleta de requisições e transmissão de mensagens entre aplicações e os *Data Integrator Job Server e Engine.*
- *Data Integrator Designer:* esse componente é dotado de uma interface gráfica que permite criar, testar e executar *jobs.* Somado a isso, também possibilita a criação e gerenciamento de mapeamento de dados, transformações, objetos, fluxos de dados, entre outros. Os quais são armazenados em um *Data Integrator Repository*.
- *Data Integrator Engine*: utilizado para a execução de tarefas (armazenadas em um *Data Integrator Job Server*) relacionadas à extração, transformação e movimentação de dados.
- *Data Integrator Job Server*: é utilizado para iniciar tarefas (movimentação de dados, integração com bases de dados, *transformations*) que estão armazenadas em repositórios, ou executar requisições oriundas do *Data Integrator Access Server.*
- *Data Integrator Repository*: é constituído por um conjunto de tabelas com a finalidade de armazenar definições a cerca de objetos, *transformations*, e afins. Cada repositório é associado com no mínimo um *Data Integrator Job Server* e pode ser do tipo local ou central (multi-usuários).
- *Data Integrator Web Administrator*: é uma ferramenta web utilizada para a execução de tarefas como: gerenciamento de *jobs*, configuração de serviços (e dos outros componentes), entre outras tarefas.

Conforme a Figura 5, é possível identificar a relação entre os componentes da ferramenta.

Figura 5 - Relação entre componentes da ferramenta SAP BusinessObjects Data Integrator

Fonte: *Data Integrator Core Tutorial*, (SAP, 2012) – pág: 24

Com relação a mecanismos de entrada e saída de dados, essa ferramenta possibilita a conexão mecanismos de armazenamento como: Attunity, DB2, Informix, MS SQL Server, MySQL, Netezza, Oracle, Sybase, Teradata, J.D. Edwards, Siebel, PeopleSoft, arquivos em Excel, arquivos texto, entre outros, conforme a documentação *Data Integrator Getting Started Guide* (SAP, 2012).

Além dos mecanismos acima citados, essa ferramenta também possibilita a conexão com serviços da nuvem, conforme Taurion (TAURION, 2009) (a exemplo da SalesForce) e *web services*, os quais ampliam os recursos disponíveispara etapas do *workflow* com o meio externo.

Outra característica é o fato da ferramenta prover mecanismos de auditoria a cerca da utilização dos dados, marcação de indicadores ao longo do processo, criação de perfis de dados (*data profile*) associados à medição de qualidade dos dados e mecanismos de recuperação de tarefas em caso de problemas ocorridos ao longo do processo.

3.6 Microsoft SQL Server Integration Services

O Microsoft SQL Server Integration Services (SSIS) surgiu como uma das ferramentas oferecidas no portfólio do Microsoft SQL Server a partir da versão 2005, após uma remodelação da então ferramenta DTS (*Data Transformation Services*). Essa ferramenta

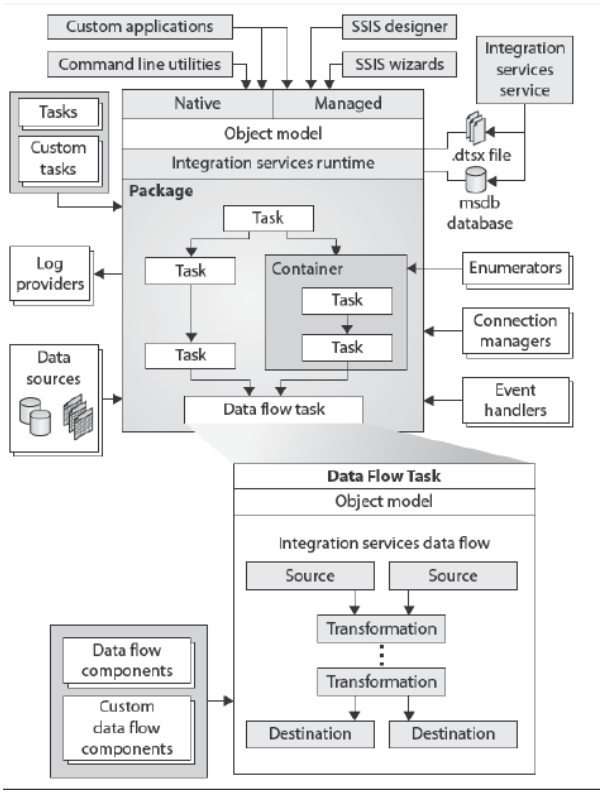
é composta por utilitários com a finalidade de prover mecanismos para importação e exportação de dados, tarefas de ETL, limpeza de dados, geração de relatórios, entre outros.

No que se refere à arquitetura, a mesma é composta principalmente pelos seguintes componentes (NANDA, 2011):

- *Integration Services Data Flow Engine:* um *dataflow* diz respeito a tarefas de extração, manipulação e carregamento de dados. Os mesmos são compostos por partes como *source*, que são utilizados quando os dados são acessados a partir de uma origem. Além do *source,* existem os *transformations*, que são fluxos utilizados para a transformação de dados, e os *destinations*, que são utilizados para a entrega (destino) dos dados. Logo, esse mecanismo é utilizado para gerenciar os componentes de um *dataflow*.
- *Integration Services Service*: é o mecanismo que da o suporte para a execução de pacotes, monitoramento de execução, conexão de múltiplos *Integration Services Servers*, entre outros.
- *Integration Services Object Model*: é utilizado para a escrita de componentes como *tasks* ou *transformations* em uma linguagem compatível com a CLR (*Common Language Runtime*). Através desse mecanismo, podem ser escritos, pacotes customizados e pode ser facilitada as tarefas de manutenção dos mesmos.
- *Integration Services Run-time Engine*: esse mecanismo possibilita a manipulação de pacotes, *tasks* e containers em tempo de execução. Somado a isso, também são possibilitados serviços como *breakpoints*, geração de logs, manipulação de conexões, entre outros.

Conforme a Figura 6 é possível analisar a arquitetura da ferramenta, e com isso, poder distinguir seus principais componentes e relação entre os mesmos.

Figura 6 - Arquitetura da ferramenta SSIS



Fonte: (NANDA, 2011) - pág.: 18.

Com relação à interfaces de desenvolvimento de *workflows* essa ferramenta é integrada ao Visual Studio, mas também possibilita a utilização do SSIS Designer, Query Builder, entre outros. Além disso, possui alguns utilitários ("*wizards*") como: SQL Server Import and Export Wizard, Integration Services Connections Project Wizard, Package Installation Wizard, Package Configuration Wizard.

No que se refere à entrada e saída de dados a ferramenta possui conexões como ODBC, OLE DB, HTTP, FTP, Arquivos Flat, SAP BI, entre outros, conforme Microsoft (2012).

3.7 Talend Open Studio & Integration Suite

A ferramenta Talend Open Studio, pertence à empresa Talend e foi lançada no mercado no ano de 2005, como uma das primeiras alternativas o*pen source*. Como ferramenta com funcionalidades que podem auxiliar em projetos de ETL, BI e afins.

Quanto a estrutura, a mesma foi construída com base na linguagem Java, e através da mesma é possível realizar tarefas relacionadas a replicação, sincronismo, integração de dados,

a análise e medição da qualidade, entre outras necessidades enfrentadas em processos de integração.

Possui uma arquitetura modular composta por mais de 400 componentes, os quais podem ser utilizados para o gerenciamento de etapas, transformações, e validações de processos ETL, armazenamento de metadados, comunicação com recursos externos ao processo (*web services*, depósitos de dados, e afins), entre outros (LACY, 2012).

Somado a isso, a ferramenta possui a possibilidade de geração de código em Java, Perl ou SQL e com isso não necessita de uma ambiente baseado em um servidor de execução de processos.

Dentre outras funcionalidades, a mesma possibilita a definição de permissões de usuários quanto a criação de objetos dentro de um processo de integração. Além disso, a ferramenta permite a importação e exportação de projetos, geração de documentação, execução individual de partes do processo (ex.: testes de validação de determinados *jobs*), entre outras (TALEND, 2012).

No que se refere a comunicação com sistemas de gerência para serem utilizadas na extração ou carga de dados, a ferramenta é compatível com ferramentas como: DB2, Firebird, Ingress, Interbase, Informix, MySQL, Microsoft SQL Server, Oracle, PostgreSQL, SalesForce, SAP, Sybase, SQLite, Teradata, entre outros.

3.8 Pentaho Data Integration (PDI)

Quanto a ferramenta Pentaho Data Integration, também conhecida como Kettle ou PDI a mesma começou a ser construída em meados dos anos 2000, quando Matt Casters, um dos fundadores do Kettle Project, perante a problemas com ferramentas de integração, iniciou as atividades referentes à criação de uma nova ferramenta que pudesse prover recursos de ETL (CASTERS, 2010).

Atualmente o PDI, é uma das ferramentas que compõem o Pentaho BI Suite, o qual é composto além do PDI, por um conjunto formado principalmente pelas seguintes ferramentas (ROLDÁN, 2010):

- *Analysis*: através do Mondrian OLAP essa funcionalidade é baseada em um servidor que prove mecanismos de exploração de relatórios.
- *Reporting*: mecanismo que permite a criação e a distribuição de relatórios em formatos como HTML, PDF, entre outros.

- *Data Mining*: através do Weka Project (WEKA, 2012) essa funcionalidade possibilita a execução de algoritmos com vistas à descoberta de novos conhecimentos com base nos dados envolvidos, de acordo com os conceitos de *Data Mining* (IMMON, 2002).
- *Dashboards*: esse recurso é utilizado para monitor os processos de acordo com determinados indicadores pré-estabelecidos (KPI). E com isso, as informações a cerca desses indicadores podem ser transmitidas de maneira gráfica para o usuário.

Somado a essas funcionalidades da suíte Pentaho BI, é importante destacar que o PDI pode ser utilizado como um mecanismo que provê soluções para as tarefas de ETL em projetos de integração.

Como faz parte da suíte Pentaho BI, o PDI pode ser utilizado em conjunto com as outras ferramentas, cada uma sendo executado de acordo com as demandas do projeto (ex.: o PDI é utilizado para o processo de ETL, o Pentaho *Data Mining* para a posterior manipulação dos dados, e o *Reporting, Analysis e o Dashboards* para acompanhar e enviar dados para um destino a cerca do andamento do workflow, entre outras formas).

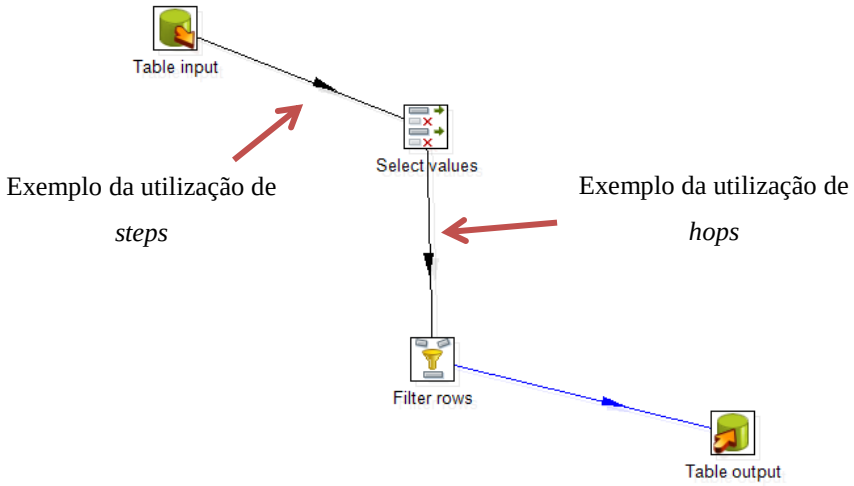
No que se refere à organização de componentes, o PDI é composto principalmente pelas seguintes aplicações:

- Carte: servidor que execução de tarefas através de um cliente remoto.
- Kitchen: ferramenta utilizada para a execução via linha de comando de *jobs* construídos através do Spoon.
- Spoon: é uma ferramenta utilizada como interface gráfica para a modelagem e construção de *jobs* e *transformations a* serem executados pelo PDI.
- Pan: ferramenta utilizada para a execução via linha de comando de *transformations* construídas através do Spoon.

Quanto aos componentes citados anteriormente, nesse estudo é importante dar ênfase para a ferramenta Spoon, considerando que a mesma possibilita a execução a criação, teste e execução de *workflows* de ETL para projetos de integração através de uma interfacegráfica.

Dessa forma, no que se refere ao Spoon, ao desenvolver um *workflow* (conforme a Figura 7) é importante ressaltar que o mesmo é composto por *steps*, que são unidades mínimas de uma *transformation,* concebidas como etapas detentoras de uma função no *workflow* (ex.: entrada de dados, conversão de valores, saída, entre outros), ligadas através de *hops*(representação gráfica, que permite o fluxo de dados entre os *steps*).

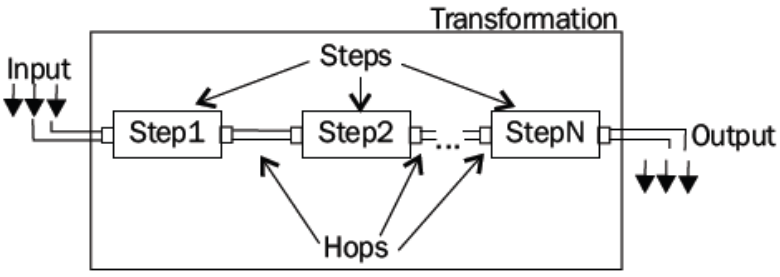
Figura 7 - Exemplo da composição de um *workflow* hipotético construído no Spoon.



Ao construir um *workflow* de integração, é relevante fazer a distinção entre o conceito de criação de *Jobs* e o de *Transformations*:

- *Jobs*: dentro do escopo de conceitos do PDI e do Spoon, os *Jobs* podem ser comparados a um processo. Logo é uma entidade criada para a execução de processos como manipulação de arquivos, validações, esses que são constituídos por *transformations* e *steps.*
- *Transformations*: são entidades formadas pelos *steps* ligados através de *hops*, e com isso são utilizadas na manipulação do fluxo de dados em um *workflow*. Um conjunto de *transformations* pode compor um *job*,de acordo com a Figura 8.

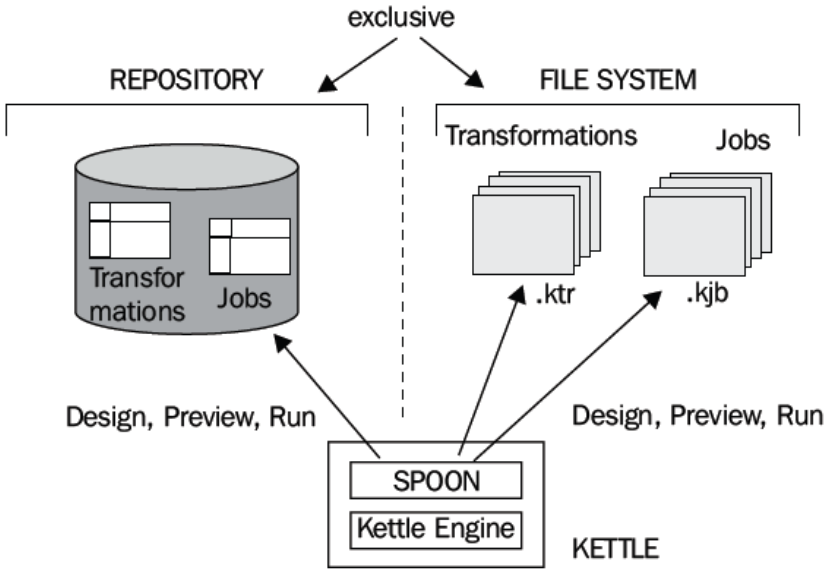
Figura 8 - Exemplo genérico de uma *transformation*.



Fonte: (ROLDÁN, 2010) - pág. 25.

Com isso em mente, após a criação de *jobs* e *transformations*, os mesmos podem ser armazenados em um repositório, ou em um arquivo, e a partir disso o *workflow* de extração, transformação e carga dos dados pode ser executado.

Figura 9 - Formas de salvamento de um *workflow* e mecanismo de execução.



Fonte: (ROLDÁN, 2010) - pág. 19.

Com relação a recursos, a ferramenta PDI possibilita a inserção de transformações e validações de dados baseadas em trechos de código escritos em linguagens de programação como Java e JavaScript, a integração do *workflow* de ETL com sistemas externos como ERP (*Enterprise Resource Planing*) ou CRM (*Customer Relationship Management*), serviços de *cloud* (TAURION, 2009), entre outros. Além disso, existem mecanismos de auditoria e medição de qualidade dos dados manipulados ao longo do processo, também existem funcionalidades relacionadas a integração em tempo real – síncrona, baseada em recursos como JMS (*Java Message Service*), mecanismos de captura de eventos nas origens CDC, entre outros.

Logo, quanto aos drivers suportados para conexões de entrada e saída de dados, podem ser citados (PENTAHO, 2012): Oracle, MySQL, AS/400, MS Access, Ms SQL Server, IBM DB2, PostgreSQL, Intersystems, Sybase, Gupta SQL Base, Dbase III, IV or 5.0, Firebird SQL, Hypersonic, MaxDB (SAP DB), Ingres, Borland Interbase, ExtenDB, Teradata, Oracle RDB, H2, Netezza, IBM Universe, SQLite, Apache Derby, Generic, além de drivers de conexão com a nuvem como SalesForce, entre outros.

3.9 Comparação

Após realizar uma análise a cerca de determinadas ferramentas é importante comparar as mesmas através de uma tabela. Para cada ferramenta será atribuída uma letra, e

para cada critério estabelecido ao longo do estudo, será atribuído um valor referente ao mesmo, com a finalidade de indicar se o mesmo possui ou não determinado critério.

Essa padronização foi estabelecida com vistas à organização e melhor compreensão da comparação entre as ferramentas.

Dessa forma, a disposição das ferramentas e as letras ficaram distribuídas da seguinte forma:

Tabela 3 – Distribuição das letras correspondentes à cada ferramenta

Letra	***Ferramenta***
A	Oracle Golden Gate
B	Oracle Data Integrator
C	IBM Cognos (Data Manager)
D	Informatica PowerCenter
E	SAP BusinessObjects Data Integrator
F	Microsoft SQL Server Integration Service
G	Talend Open Studio & Integration Suíte
H	Pentaho Data Integration

Somado a isso, a disposição dos critérios e os números ficaram distribuídos de acordo com a Tabela 4. Esses critérios foram escolhidos ao longo do estudo, os quais representam pontos importantes que devem ser levados em consideração ao construir e executar *workflows* de ETL, com o auxílio de uma ferramenta com essa finalidade:

Tabela 4 – Distribuição das letras correspondentes à cada ferramenta

Letra	***Ferramenta***
1	Plataforma (de acordo com as versões disponíveis no site principal da ferramenta)
2	Tipos de Licença
3	Possibilidade de integração com outras ferramentas e serviços externos (ex. outras ferramentas de integração, *webservices*, entre outros)
4	Extração de dados de diversas origens.
5	Carga de dados em diversos formatos
6	Permite a integração com serviços na nuvem

7	Permite o paralelismo de operações
8	Permite a customização da ferramenta a nível de código (*opensource*)
9	Permite a automatização de tarefas sem a necessidade de intervenção humana
10	Permite a utilização de linguagens de programação ao longo do processo

Por fim, conforme a Tabela 5, foi criada uma tabela com vistas a comparar os critérios elencados e os respectivos valores.

Tabela 5 – Tabela comparativa entre os critérios e as ferramentas

Critérios	1	2	3	4	5	6	7	8	9	10
A	Windows, Unix-derivedsystems	Proprietária	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim
B	Windows, Unix-derived systems.	Proprietária	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim
C	Windows, Unix-derived systems.	Proprietária	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim
D	Windows, Unix-derived systems.	Proprietária	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim
E	Windows, Unix-derived systems.	Proprietária	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim
F	Windows.	Proprietária	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim
G	Windows, Mac OS, Unix-derived systems.	GPL/Apache License	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
H	Windows, Mac OS, Unix-derived systems.	Apache License	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim

Após a tabela comparativa, é possível constatar que todas as ferramentas analisadas fornecem os principais recursos quanto a tarefas envolvidas em processos de ETL como: paralelização (independente da velocidade), tarefas de automação, monitoramento, geração de relatórios, comunicação com serviços externos, entre outros.

No entanto, existem particularidades quanto as interfaces gráficas, desempenho, plataforma, modelos de licenciamento, fontes de dados utilizadas para extração ou carga de dados, características que apesar de semelhantes, não são idênticas em todas as ferramentas.

Com relação às fontes de dados, não foi possível detectar com a mesma precisão nas outras ferramentas, quais as fontes de dados para a ferramenta Microsoft SQL Server Integration Services. Pois na análise da mesma, foram encontrados os pacotes de gerenciamento de conexões, ao invés das fontes de dados específicas.

No que se refere a versões, a maior parte das ferramentas de ETL analisadas, são distribuídas em versões diferentes, no entanto a ferramenta PowerCenter é uma ferramenta que possui versões, mas também edições diferentes como*: Standard, Advanced, Data Virtualization, Real Time e Cloud Edition,* as quais apresentam variações quanto a funcionalidades. Uma justificativa para esse ponto, pode ser o fato de a maior parte das ferramentas analisadas fazerem parte e serem distribuídas como componentes de ETL para suítes de BI, ao invés da PowerCenter que é uma suíte de ETL.

Com relação às ferramentas proprietárias e as ferramentas *open source*, de acordo com o estudo realizado, foi possível inferir que as seis ferramentas proprietárias analisadas apresentam uma melhor preparação para contextos que exigem a manipulação de um maior volume de dados. Com isso necessitando de uma arquitetura escalável capaz de fornecer maior processamento conforme a demanda, aliado aos mecanismos de monitoramento e prevenção de falhas ao longo do processo.

Quanto ao critério de customização de código (*open source*) é importante destacar que na implementação do estudo de caso, esse recurso não apresenta uma grande importância. No entanto, é um fator que pode ser significativo para outros contextos, onde a personalização da ferramenta seja necessária.

No que se refere à documentação, é importante destacar que as ferramentas da Oracle (ODI e GoldenGate), Microsoft (SSIS), SAP (BusinessObjects Data Integrator) apresentam arquivos com facil disponibilidade em seu site, além de possuírem no mínimo um livro publicado por editora para as respectivas ferramentas. Por outro lado, a ferramenta da IBM (Cognos Data Manager) e também a da SAP, por fazerem parte de uma suíte de BI, apresentam na maior parte de suas documentações, especificações voltadas para a suíte de BI, e não com uma ênfase específica no componente de ETL. Nesse ponto a ferramenta da Informatica (PowerCenter), apresenta documentações disponíveis, mas com a necessidade de cadastramento e outras particularidades que acarretaram em uma maior dificuldade na procura por manuais técnicos no site da empresa.

Lembrando que para todas as ferramentas analisadas existem documentações de terceiros (blogs, sites pessoais e afins), os quais podem auxiliar no conhecimento de mundo dos usuários das ferramentas. Fator esse que é bastante notável e diferenciado de maneira positiva quanto as ferramentas PDI e a Talend Open Studio & Integration Suite. As quais apresentam documentações disponíveis em seus sites oficiais, além de livros publicados em editoras, comunidades de desenvolvedores e usuários, fóruns e outros fatores que facilitam a disseminação de informações. Isso com um papel mais perceptível na ferramenta Pentaho Data Integration (PDI).

Por fim, ao longo da análise de ferramentas integração, foi possível constatar as seguintes características a cerca da Pentaho BI Suíte, em especial ao PDI, como justificativa para a utilização dessa ferramenta na tarefa de ETL, aplicada ao estudo de caso:

- Formas de licenciamento e utilização da ferramenta: no que se refere às formas de licenciamento, conforme Pentaho (PENTAHO, 2012), existe uma alternativa proprietária da Pentaho Corporation (*Enterprise Edition*) com um número maior de funcionalidades disponíveis, somado ao suporte mais amplo quanto à sua utilização. Mas também existem outras possibilidades que podem ser utilizadas na versão *Community Edition* de acordo com o padrão GPL, *Apache License* (tipo de licença do PDI), *Mozilla Public License* entre outras formas. Esse item foi um fator importante da escolha dessa ferramenta, considerando que de acordo com esse contexto, não é necessário o pagamento de licenças de utilização.
- Colaboração entre usuários e desenvolvedores: é bastante notável a rede de colaboração a nível mundial entre usuários e desenvolvedores no que tange à suíte Pentaho BI, a exemplo da Pentaho *Community* (PENTAHO, 2012), e ao Pentaho *Kettle Project* (PENTAHO, 2012), e outros projetos que compõe o conjunto. Pois através desses espaços é possível compartilhar experiências, auxiliando outros usuários e também buscar auxílio com outras pessoas.
- Documentação disponível: Com relação a esse ponto foram encontrados no site da *Amazon* uma quantia equivalente a quatro livros diretamente relacionados com a ferramenta, além de outros que citam o nome da ferramenta. Somado a isso, existem o site da Pentaho, a exemplo do Pentaho Fórum (PENTAHO, 2012), blogs e afins onde os usuários e desenvolvedores trocam informações. Isso pode ser considerado um ponto a favor, pois no que se refere a essa ferramenta existem documentos de autorias distintas, e não somente documentos criados sob o escopo da Pentaho *Corporation*.

- Ferramenta *open source*: essa vantagem dentro da versão Community Edition, viabiliza que a ferramenta seja modificada e customizada, a nível de código, de acordo com a necessidade de seus utilizadores.

4 ESTUDO DE CASO

O cenário escolhido para o desenvolvimento do estudo de caso consiste em uma empresa de desenvolvimento de software existente a mais de cinco anos, composta por mais de cinquenta funcionários e que atua no desenvolvimento de soluções para clientes em inúmeros seguimentos como: aplicações web, portais corporativos, intranet, extranet, mobile, CRM, ERP, BI, integração de dados oriundos de sistema e esquemas de armazenamento heterogêneos, entre outros.

Essa empresa possui equipes dispersas geograficamente através de suas filiais localizadas em diversas cidades, distribuídas em mais de um país. As equipes são compostas por vários membros, que trabalham em atividades de consultoria a clientes, desenvolvimento de novas aplicações ou manutenção de aplicações.

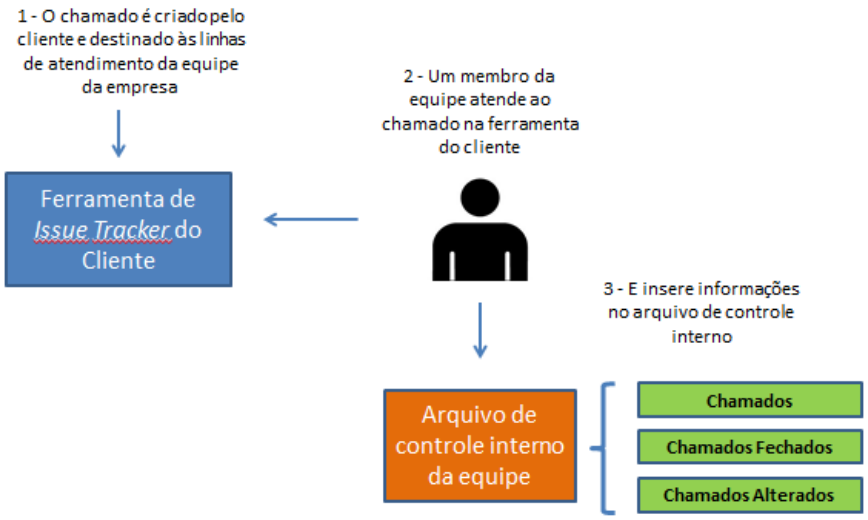
Uma dessas equipes de manutenção trabalha no atendimento a chamados de defeitos referentes a aplicações utilizadas por um determinado cliente, o qual é detentor de um número equivalente a cinco aplicações de grande porte. Esses defeitos são reportados e administrados através de uma ferramenta proprietária do cliente *(Sistema de Rastreamento de Defeitos)*, entretanto para fins de controle interno da equipe de manutenção, esses defeitos são inseridos em um arquivo de controle próprio da empresa.

A proposta deste estudo de caso, consiste na elaboração de uma alternativa que visa a migração do esquema de armazenamento do arquivo de controle interno da equipe, para um modelo que baseia-se em uma base de dados relacional.

4.1 Análise de Contexto

Nesse contexto, quando um membro da equipe atende um chamado na ferramenta do cliente, paralelamente, o mesmo deve inserir um registro em um arquivo de controle interno, que é composto por três planilhas armazenadas na ferramenta Google Docs (GOOGLE, 2012), conforme a Figura 10.

Figura 10 - Figura demonstrativa do procedimento de atendimento aos chamados

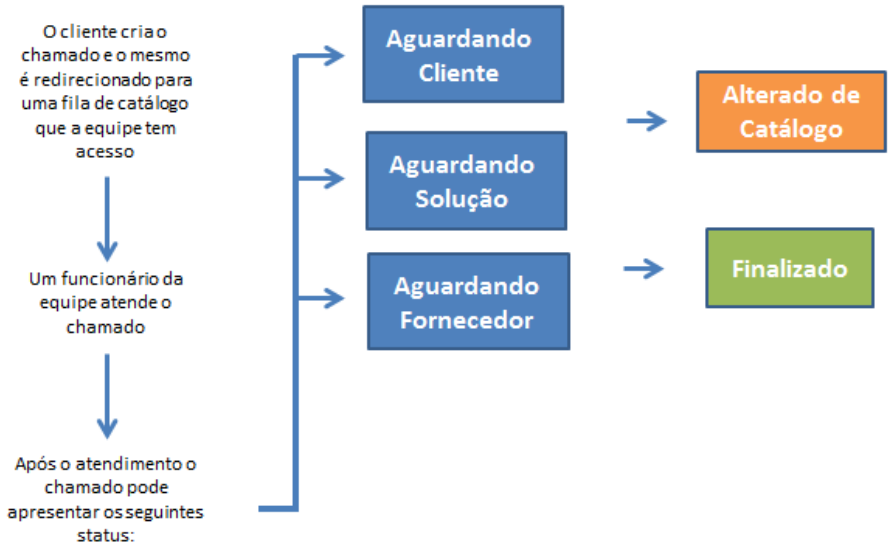


A Figura 10, representa o procedimento de atendimento aos chamados. Somado a isso, existe um procedimento referente ao desenvolvimento da solução para o problema reportado no chamado, pois ao atendê-lo, o funcionário pode alterar o chamado de catálogo, finalizá-lo ou designá-lo aos seguintes status:

- Aguardando cliente: nesse caso o funcionário solicita para quem criou o chamado, maiores informações a cerca do problema ocorrido.
- Aguardando solução: esse status é o destino de chamados, onde a aplicação em questão já passou por alteração de código. Portanto, basta essa alteração entrar em produção e o problema será resolvido.
- Aguardando fornecedor: esse status é utilizado quando o funcionário necessita da intervenção do cliente para a resolução do problema (ex.: documentações, dados os quais o funcionário não tem acesso, entre outros).

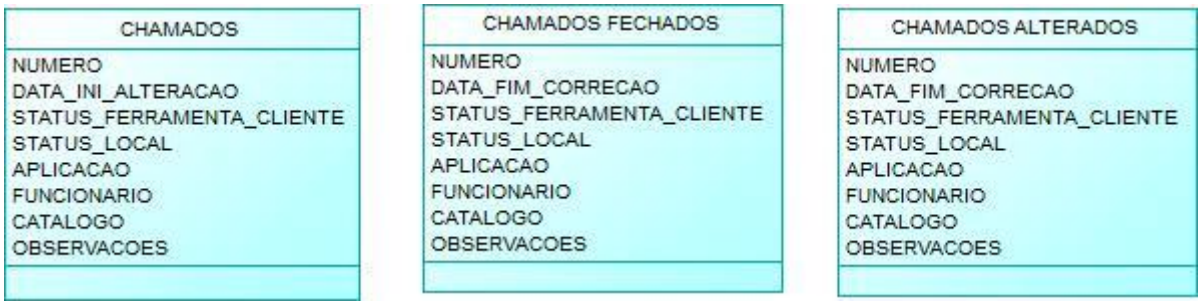
Após os status anteriores, o chamado pode ser alterado de catálogo (destinado a uma outra fila de atendimento referente problemas específicos), ou finalizar o chamado com uma respectiva solução para o problema, conforme a Figura 11.

Figura 11 - Status que um chamado pode assumir após ser atendido.



Com relação ao arquivo de controle, o qual é utilizado para fins de controle interno pela equipe de manutenção de maneira paralela com a ferramenta de rastreamento de defeitos do cliente, o mesmo é composto por três planilhas dispostas conforme a Figura 12.

Figura 12 - Figura demonstrativa das planilhas e as respectivas colunas.



A planilha *Chamados* tem a finalidade de armazenar chamados que foram recentemente atendidos e que não foram solucionados ou alterados de catálogo.

Somado a isso, a planilha *Chamados Fechados* tem a função de armazenar chamados que já passaram pela equipe de atendimento da empresa, e que por sua vez já foram finalizados. Apresenta um número aproximado de dois mil registros de chamados atendidos, em cinco aplicações, as quais a empresa de software presta assistência para o determinado cliente.

Por fim, a planilha *Chamados Alterados* tem a função de armazenar chamados que foram destinados para as filas de atendimento da equipe. Entretanto, os mesmos não dizem respeito a problemas que a equipe possa resolver. Por isso esses são destinados para outra

linha de catálogo, também chamado de fila de atendimento. No que se refere à quantidade de registros, essa planilha também apresenta um número aproximado de dois mil registros.

4.2 Problema

Como relatado na seção anterior, o gerenciamento dos chamados é realizado por uma ferramenta proprietária mantida pelo cliente, na qual os funcionários da equipe da empresa de software possuem um nome de usuário e uma senha. Com isso, acessam a ferramenta e verificam chamados para serem atendidos nas filas de problemas das aplicações, alteram o status e resolvem o problema de chamados ou encaminham para outras filas de atendimento.

Dessa forma, por mais que a ferramenta do cliente atenda às necessidades a cerca da resolução de problemas, os funcionários também têm a tarefa de após atenderem um chamado, inserir registros a cerca do andamento do mesmo no arquivo de controle interno.

Considerando que a empresa não possui um mecanismo de rastreamento de falhas, para essa equipe de manutenção, esse arquivo de controle foi criado com o objetivo de gerenciar as atividades dos membros da equipe, os defeitos solucionados, geração de relatórios (o que a ferramenta de rastreamento de defeitos do cliente não possibilita) entre outros.

Por isso, foi criado o arquivo de controle interno baseado em arquivo composto por três planilhas no serviço Google Docs (GOOGLE, 2012), onde a finalidades do mesmo é a de possibilitar aos membros da equipe e coordenadores do andamento das atividades em inúmeros tópicos. Dentre esses tópicos, podem ser destacados: (I) as aplicações que estão apresentando problemas, (II) as funcionalidades, (III) quantidade de incidentes ocorridos, (IV) o caminho de resolução de um chamado, (V) quais chamados foram atendidos por um funcionário, entre outros.

Logo, no que se refere à adoção do arquivo de controle em um formato composto por planilhas compartilhadas, podem ser destacados alguns pontos positivos, conforme os descritos abaixo:

- No momento em que os funcionários detectam um novo chamado para ser atendido nas filas da equipe, é simples a criação de um registro na planilha Chamados. Isso sendo proveitoso, considerando que basta preencher poucos campos com tipos de dados simples, evitando o desperdício de tempo do funcionário em uma atividade burocrática.

- Considerando que o arquivo está hospedado na nuvem (TAURION, 2009), o gestor da equipe pode compartilhar o arquivo com os seus funcionários, e com isso, os mesmos podem acompanhar em tempo real o histórico de um determinado chamado. Podendo acessar o arquivo de qualquer lugar que tenha conexão com a internet, inclusive externamente à rede da empresa.
- O modelo adotado herda as vantagens de armazenar informações na nuvem (TAURION, 2009): redundância, escalabilidade, disponibilidade, entre outras.

Por outro lado, o modelo adotado apresenta problemas dentre os quais, alguns exemplos seguem abaixo:

- A adoção de planilhas como forma de armazenamento das informações ao invés de um banco de dados relacional, aliado ao descuido ao preencher informações, implicou em sérios problemas de restrição de integridade como: registros repetidos, datas inválidas, preenchimentos de campos com formato inválidos, campos de importância com valor nulo ou vazio, e afins.
- Caso algum membro da equipe esteja mal intencionado, ou até mesmo por descuido, inúmeros dados podem ser alterados ou excluídos sem controle ou restrição alguma, devido a não existir um mecanismo de controle de acesso. Sem falar na dificuldade de implantação de mecanismos de auditoria (o que em um banco relacional pode ser implementado através do gerenciamento de permissões, gatilhos, ou outras formas).
- O fato de as planilhas estarem em um arquivo compartilhado em uma nuvem fora da empresa, acarreta em problemas de indisponibilidade nos momentos em que os funcionários não conseguem acessar a conta de e-mail.
- O modelo de planilhas adotado, demanda um grande esforço para permitir a junção de informações para responder consultas a cerca de estatísticas do andamento das atividades.
- O número elevado de registros ocasionados por um mau planejamento do recurso na nuvem, fez com que a utilização do arquivo se tornasse um procedimento lento. Muitas vezes ocasionando em erros na inserção, alteração ou exclusão de dados. O que passou a preocupar os membros da equipe quanto ao futuro dessa prática adotada.

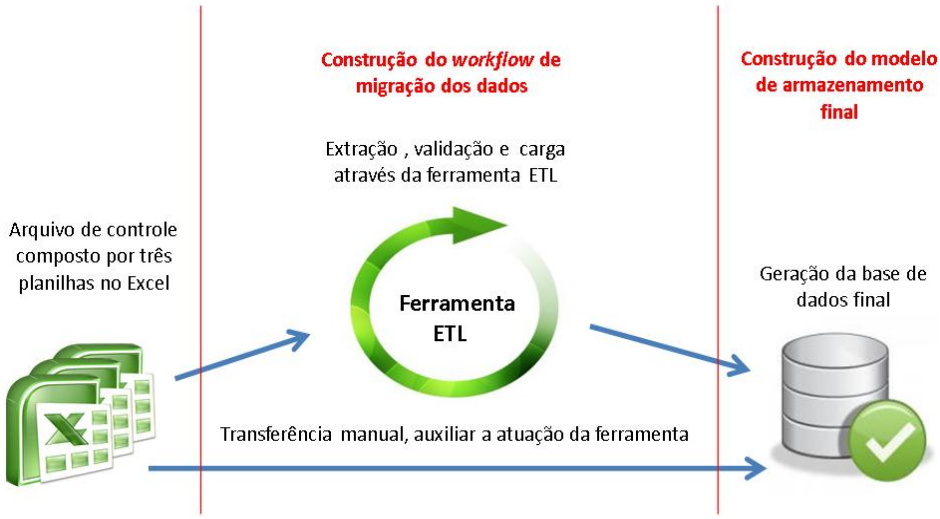
Logo, de acordo com a exposição do contexto acima relatado, é possível buscar critérios para as etapas de planejamento e implementação da alternativa de armazenamento que deve

auxiliar a empresa na resoluçãode problemas enfrentados pela estratégia anteriormente adotada.

4.3 Implementação

No que se refere à implementação do estudo de caso é importante ressaltar que a mesma baseou-se no desenvolvimento de uma sequência de etapas. Esse conjunto de etapas é composto pela etapa de análise do contexto inicial, o levantamento de requisitos com base nos problemas apresentados e a modelagem da base de dados destino. O resultado desse processo foia construção do fluxo de ETL utilizado para o transporte e realocação dos dados entre a origem e o destino, conforme a Figura 13.

Figura 13 - Figura demonstrativa das subdivisões da implementação.



Nessa ideia, é importante justificar que as etapas de análise do contexto e levantamento de requisitos têm como base os itens 4.1 e 4.2 desse estudo. Logo, nos itens subsequentes serão descritos as etapas de construção do esquema de armazenamento, e a construção do *workflow* de migração dos dados.

4.3.1 Construção do modelo de armazenamento final

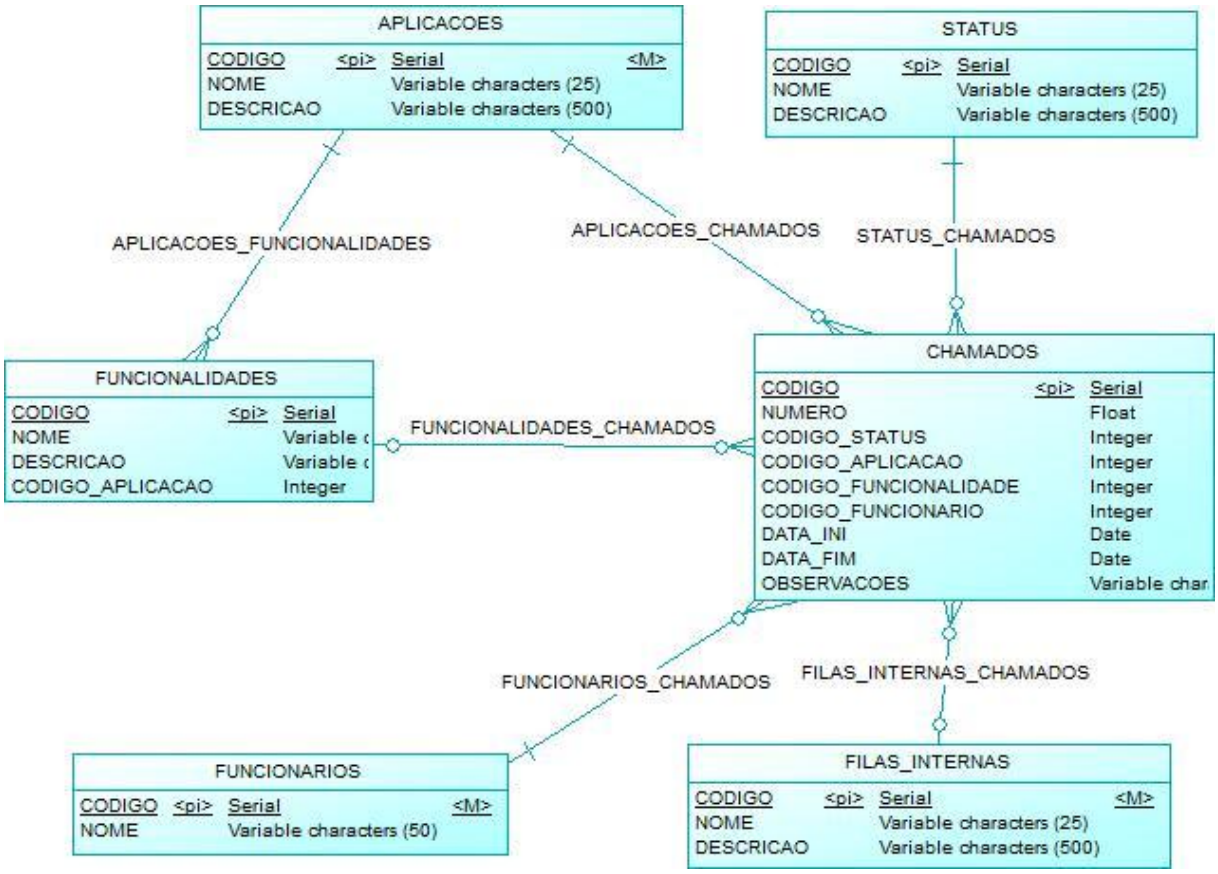
Para a construção da modelagem da base de dados, responsável por armazenar os dados ao final do processo de migração, foi levado em conta a análise do esquema de armazenamento utilizado no arquivo de controle, somado ao levantamento de informações referentes às necessidades presentes e futuras quanto aos dados alocados nesse esquema.

Esse novo modelo (de acordo com a Figura 14), visa através de uma visão *bottom-up*, contemplar um armazenamento otimizado (no que se refere a itens como redundância de

informações, preservação de dependências, desempenho de transações e escalabilidade) ao comparar com o modelo anterior das planilhas utilizadas no arquivo de controle.

Outro ponto importante a ser destacado é a necessidade de realocar os dados em um modelo simplificado que permita, caso necessário, futuros ajustes e sua estrutura. Como exemplo disso, pode ser destacada a possibilidade de estudos futuros voltados a uma futura utilização dos dados, alocados no modelo criado por um sistema de rastreamento de defeitos, ou até mesmo para práticas de mineração de dados (IMMON, 2002), entre outros.

Figura 14 - Figura demonstrativa do MER da base de dados final.



Dessa forma, após estar ciente do modelo anterior e da modelagem final prevista para o novo sistema de armazenamento, é importante levar em consideração alguns tópicos para fins de comparação entre a origem e o destino dos dados.

Nessa ideia, as tabelas *aplicacoes*, *funcionarios* e *status* surgiram como alternativa para problemas como a redundância de informações existentes em colunas nas três planilhas. Além disso, os dados que antigamente eram do tipo String (status_ferramenta_cliente, status_local, aplicacao, funcionario), após a conversão para o formato de alocação das tabelas, possibilitou a utilização de relacionamentos com a tabela chamados. Com isso, as informações que estavam replicadas nas três planilhas, foram filtradas e realocadas em tabelas

específicas. Podendo dessa forma, serem referenciadas na tabela chamados, preservando a dependência entre os registros. Possibilidade essa que é vital para questões como: integridade referencial, redundância de informações e desempenho em transações.
Por outro lado, a tabela filas_internas é uma tabela que no momento do processo de integração não vai ser utilizada. Isso devido ao fato de que as antigas planilhas não possuirem informações referentes à fila de atendimento em que o chamado foi destinado para a equipe atender. Logo, para o provimento dessa informação, se possível, seria necessária a utilização de técnicas relacionadas ao conceito de imputação de dados (CASTANEDA et. al., 2008).

Entretanto, para os novos chamados que forem armazenados no sistema, considerando que ao cadastrar cada chamado, o membro da equipe terá acesso a informação de qual fila de atendimento interna o chamado está localizado, será possível reter essa informação. Devido a isso, a tabela foi criada, com vistas a prover mais uma possibilidade quanto a utilização do novo sistema.

Somado a isso, durante a elaboração do projeto, constatou-se a necessidade de armazenar informações a cerca da(s) funcionalidade(s) das aplicações que diz respeito ao(s) chamado(s). No entanto, nas planilhas antigas, existem somente os registros de qual aplicação é impactada por um chamado, e não qual a funcionalidade pertencente à aplicação é que está apresentando o problema.

Perante a esse cenário optou-se pela criação das relações: aplicacoes_chamados, funcionalidades_chamados e aplicacoes_funcionalidades. Como justificativa para isso, leva-se em consideração que ao inserir o chamado, se o mesmo for oriundo das planilhas, o mesmo não terá funcionalidade, mas para novas inserções o funcionário poderá inserir uma funcionalidade para o chamado.

Com a modelagem escolhida, optou-se pela adoção do SGBD PostgreSQL, versão 9.1, como o sistema responsável pela base de dados.No entanto, é importante ressaltar que a ferramenta utilizada para o processo de ETL, é passível de conexão com diversas fontes de dados. Logo, isso abre precedentes para que outros SGBDs possam ser utilizados para a resolução do problema de armazenamento envolvido no estudo de caso.

4.3.2 Construção do *workflow* de migração dos dados

Tendo em mente o entendimento dos requisitos esperados quanto às necessidades da empresa para o esquema final de armazenamento, seguido da criação da modelagem da base

de dados e escolha do respectivo SGBD, é que foi obtido o aporte necessário de informações para a construção do processo de ETL.

Nesse contexto, para a extração dos dados da origem, transformação e posterior carga dos mesmos no destino, foi escolhida a ferramenta Pentaho Data Integration (PDI, também chamado de Kettle), versão 4.2.de acordo com as justificativas abordadas na seção 3.3.

No que se refere à instalação e configuração do ambiente, foi optado pela criação de uma ambiente de homologação para a construção do *workflow* de ETL, simulação do ambiente final de armazenamento, testes e validação do processo. Após a validação das etapas a serem executadas, e da simulação dos resultados é que foi aplicado o *workflow* de maneira efetiva no ambiente de produção.

Para a construção e execução desses dois ambientes, foi instalada e configurada uma *workstation* com o *Java Runtime Environment* posterior a versão 1.5, juntamente com as ferramentas que compõem o PDI (descritas na seção 3.2).

Após a *workstation* estar configurada, tendo como base o MER da Figura 14, foi criada, dentro do *workflow* de ETL, uma *transformation* responsável por executar o *script* de SQL para a criação da base de dados final, como etapa antecessora à execução das *transformations* responsáveis por extrair, transformar e carregar os dados no contexto de armazenamento final das planilhas *Chamados, Chamados Fechado*s e *Chamados Alterados*.

Decisão que foi tomada com vistas a definir a etapa de criação da base de dados final dentro do PDI considerando que o script de SQL criado, é compatível com outros SGBDs, e com isso, bastaria alterar detalhes da conexão, e a base de dados poderia ser criada em um outro SGBD.

Com vistas a detectar formas de simplificar e otimizar a construção e execução do *workflow* da ferramenta, foi realizada uma análise a cerca de dados que poderiam ser transferidos fora do *workflow* construído no escopo do PDI. Nessa ideia, foi tomada a atitude de pesquisar nas planilhas *Chamados*, *Chamados Fechados* e *Chamados Alterados* os dados das colunas *funcionários*, *status_ferramenta_cliente*, *status_local*, *aplicacao, funcionario e catalogo* e tomar as seguintes iniciativas:

- Devido ao pouco número de funcionários existentes na equipe, foi optado por detectar quais os funcionários existentes e com isso popular manualmente a tabela FUNCIONARIOS. Dessa forma cada funcionário terá o campo codigo (uma chave primária da tabela FUNCIONARIOS) mapeado no *workflow* de carga na tabela CHAMADOS.

- Detectar quais asaplicações existentese com isso popular manualmente a tabela APLICACOES. Dessa forma cada aplicação terá o campo codigo (uma chave primária da tabela APLICACOES) mapeado no *workflow* de carga na tabela CHAMADOS.
- Detectar quais os tipos de status existentes e com isso popular manualmente a tabela STATUS. Dessa forma,cadastatus terá o campo codigo (uma chave primária da tabela STATUS) mapeado no *workflow* de carga na tabela CHAMADOS.

Como justificativa para a iniciativa anterior, foi levado em conta o fato de os dados envolvidos apresentarem um nível baixo de complexidade e de quantidade de valores. Somado a isso, o fator de existir um número aproximado de oito funcionários, cinco aplicações atendidas nos chamados e cinco tipos de status que um chamado pode ter. Logo, essa atitude (extrair, analisar e carregar os dados de uma maneira manual) possibilitou a execução de uma parte do processo evitando a criação de mais *steps*, *hops* e *transformations* dentro do *workflow*. Ocasionando em uma economia de tempo, processamento e complexidade da execução.

No entanto, é importante ressaltar que em contextos de armazenamentos complexos, essa prática pode não ser apropriada. Com isso, deveria ser criada uma *transformation* específica para consultar os dados existentes nas planilhas e com isso efetuar a carga de maneira automática.

No que se refere ao *workflow* construído para a transferência da maior parte do volume de dados, é importante destacar que para cada planilha existente no arquivo de controle inicial, foi criado através do Spoon uma *transformation* composta por quatro *steps* e três *hops*.

A exemplo da planilha *Chamados*, a *transformation* foi criada tendo no seu início um *step* chamado de Microsoft Excel Input (extração_tabela_chamados) - 15.A, o qual dentro do processo, tem a finalidade de referenciar o arquivo de controle, o conjunto dados que o compõe e os seus metadados (planilhas, colunas, linhas, tipos de dados, entre outros).

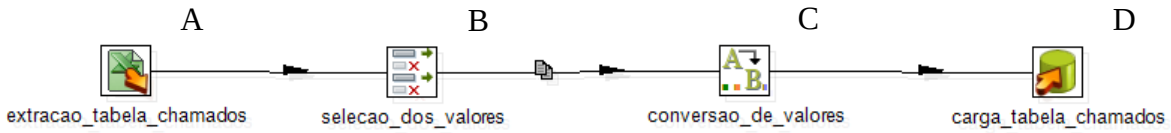
De maneira sequencial, foi inserido o *step* chamado de Select Values (selecao_dos_valores) - 15.B, que possui a tarefa de receber através de um *hop* os dados do *step* anterior e com isso renomear nomes de campos, deletar valores e estabelecer algumas validações e conversões como tamanho do campo, tipo de dado e afins.

Somado aos dois *steps* anteriores a *transformation* de cada tabela é composta pelo *step* intitulado de Replace in string (conversao_de_valores) - 15.C, que tem a função de

receber os dados, efetuar substituições de valores com base em equivalência dos dados. Esse *step*, ao receber dados como funcionário, aplicação, status_ferramenta_cliente, status_local, era incumbido de realizar verificações e transformações de valores de acordo as chaves primárias das tabelas Funcionarios, Aplicacoes e Status de acordo com os seguintes exemplos hipotéticos: para o funcionário "A" converte o valor para 1, para o funcionário "B" para 2, a aplicação "Y" para 1, o status "Z" para 4.

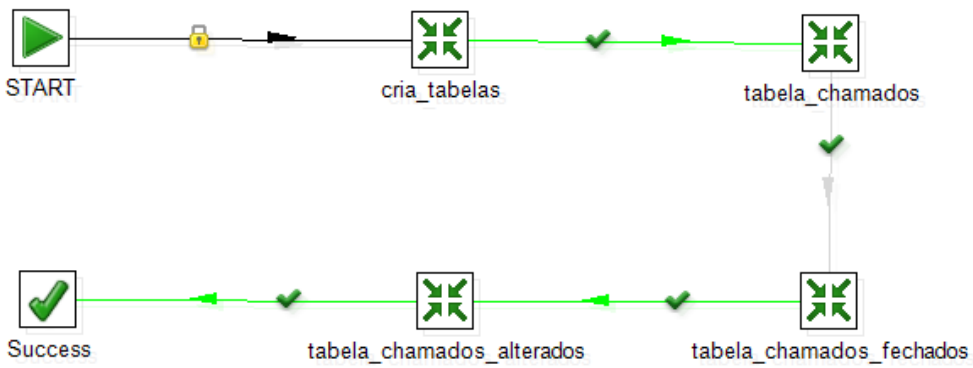
Logo, após os três passos anteriores, os dados já foram extraídos do arquivo de controle, transformados e validados, com isso, foi inserido o *step* Table output (carga_tabela_chamados) - 15.D, com a finalidade de organizar os dados recebidos e submetê-los na tabela destino de cada *transformation.*

Figura 15 - *Transfomation* referente à planilha chamados.



No entanto, a descrição da *transformation* relatada anteriormente (Figura 15) faz referencia à planilha *Chamados* existente no arquivo de controle. Vale ressaltar que o *workflow* de integração do estudo de caso é composto por um *job* que é constituído de quatro *transformations*, uma para a criação da base de dados destino, a da planilha *Chamados*, uma para a planilha *Chamados Fechados* e outra para a planilha *Chamados Alterados*, conforme a Figura 16.

Figura 16 - *job* de execução do workflow de ETL, dentro do PDI.



Entretanto, devido a semelhança entre a organização dos dados em cada planilha, as três *transformations* de cada planilha apresentam um alto grau de semelhança, o qual inclui os mesmos *steps*, com as mesmas tarefas de validação, e encaminhamento dos dados, salvo algumas particularidades.

No que se refere a particularidades de cada *transformation*, deve ser destacado a planilha origem utilizada como *input* dos dados e o campo data_ini_alteracao que na planilha chamados apresenta essa designação, diferente das outras duas planilhas onde o campo data é chamado de data_fim_alteracao.

Após a comparação das três planilhas iniciais do arquivo de controle, é possível constatar que essa diferença, não implica em grandes mudanças entre as três *transformations*.

Por fim, com base na a realização de várias execuções do *workflow* e consecutivas comparações entre os dados extraídos do arquivo de controle com os dados inseridos e buscados na base de dados destino, foi possível constatar que o *workflow* poderia serexecutado no contexto real da empresa, no ambiente de produção.

4.4 Resultados

Tendo em mente os problemas apresentados no esquema de armazenamento antigo,os quais foram parte dos pontos motivadores para esse trabalho, é possível descrever os resultados obtidos.

Nesse contexto, ao final do processo, os dados foram realocados de maneira organizada e planejada com vistas a facilitar a posterior utilização dos mesmos.

Somado a isso, através da execução de consultas à base de dados final, foi possível verificar um número relevante de registros que estavam com inconsistências em sua forma original nas planilhas do arquivo de controle. Isso devido aos problemas de integridade, e controle da manipulação dos dados existentes.

Outro ponto importante é que o fato de armazenar os dados em uma base de dados relacional, abre precedentes para uma série de vantagens como: integridade referencial, indexação de informações, escalabilidade, permissões de acesso, rotinas de backup, replicação de instâncias, entre outras.

No que se refere à geração de relatórios, devido a possibilidade de utilização da linguagem SQL, dentro da base relacional, é possível realizar junções, para geração de relatórios e também viabiliza a descoberta de novos conhecimentos.

Por fim, a implementação do estudo de caso, representado pela migração de contextos, preparou a organização dos dados para ser desenvolvida uma camada de controle dos chamados (um sistema), ou a integração com um mecanismo de rastreamento de defeitos, entre outros.

4.5 Discussão

Com relação às ferramentas de ETL analisadas, é importante ressaltar que antes do planejamento de um processo de ETL é importante ter noção das dimensões e particularidades do mesmo. E com isso em mente, poderá ser definida a escolha de uma ferramenta que possibilite a execução desse projeto.

Com relação ao exemplo abordado no estudo de caso, a ferramenta PDI atendeu às expectativas de acordo com as predefinições do problema existente e à solução planejada. No entanto, para projetos que envolvam uma maior complexidade, ou particularidades quanto a plataforma, fontes de dados, entre outros, talvez seja necessária a utilização de uma outra ferramenta que atenda as necessidades envolvidas.

Após efetuar a implementação do estudo de caso, é necessário ressaltar que ao longo da execução das tarefas referentes ao processo de migração de contextos de armazenamento, alguns pontos tiveram uma boa importância para a obtenção de êxito ao final do processo. No que se refere a esses pontos, podem ser destacados:

- A dedicação com vistas a obter o máximo de informações quanto a estrutura e as particularidades referentes ao armazenamento dos dados no contexto de origem. Somado ao levantamento dos requisitos referentes às necessidades que o novo contexto deverá suprir.
- Pesquisa e construção de práticas responsáveis por detectar de antemão possíveis falhas e problemas quanto aos valores armazenados. Isso sendo de grande valia para a criação de alternativas para, desde que possível, realizar a conversão desses valores.
- É importante destacar que devido ao nível de complexidade do estudo de caso foram utilizados alguns recursos da ferramenta PDI (ex.: *steps* de extração de dados a partir de arquivo no formato Microsoft Excel). Mas além dos utilizados, poderia ser levado em consideração possibilidades como: interação com *web services*, serviços da *cloud* (TAURION, 2009), sistemas como ERP (ex.: SAP) ou CRM, mecanismos de captura de eventos nas origens, customização de código fonte, transformações baseadas em fluxos construídos com uma linguagem de produção como Java ou outras, auditoria e medição de qualidade dos dados ao longo do processo, entre outros recursos disponíveis.

- Apesar de não efetuar testes com outras ferramentas além do PDI, é importante destacar que a interface gráfica Spoon possibilitou um bom nível de facilidade para a construção e execução do *workflow* de ETL. Considerando a facilidade para a localização dos *steps* envolvidos (tudo separado de acordo com a sua aplicabilidade, ex.: os steps de extração de dados localizados em uma guia de *Input*, assim como os de carga estão em uma guia de *Output*, e assim sucessivamente).
- A utilização de uma ferramenta de ETL pronta e já utilizada no mercado como o PDI, trouxe uma redução drástica no tempo gasto e na complexidade para construir e executar o workflow de ETL. Tendo em vista que construir uma ferramenta de maneira manual, de tal forma que oferecesse os mesmos recursos da PDI, poderia demorar mais tempo e aumentar a complexidade do processo inteiro.
- Realização de testes unitários após inserção ou alteração de *steps*, ou *tasks*, ao longo das *transformations* que compõem o *job* de manipulação dos dados. A justificativa para isso, é que ao modificar parte do processo, por mais que aparentemente não seja um ponto relevante, existe a possibilidade de causar impactos negativos no resultado final da manipulação das informações.
- A criação de um ambiente de homologação (testes) isolado do contexto de produção para a execução temporária das tarefas pode ser considerada uma boa prática no que se refere a implantação de mudanças na área de armazenamento de dados. No entanto, é necessária a ressalva de que a criação de um ambiente de simulação do contexto real, pode prevenir ações que poderiam comprometer todo o projeto.
- Após a conclusão do *workflow* de transporte dos dados entre as planilhas do arquivo de controle e a base de dados destino, o mesmo era composto por um *job* formado por quatro *transformations*. Perante a esse cenário, foi optado por executar as três *transformations* de maneira sequencial.

  No escopo do estudo de caso, devido a baixa complexidade no *workflow* construído, essa prática foi válida. Entretanto, em outros contextos, pode ser necessária a execução paralela de fluxos que compõem partes do *workflow* de manipulação dos dados.

5 CONSIDERAÇÕES FINAIS

A área de tecnologia da informação está sempre em constante modificação com vistas a oferecer um número maior de possibilidades referentes à resolução de problemas do cotidiano da sociedade. Nessa ideia, a exemplo de esquemas de armazenamento de informações, muitas vezes as alternativas adotadas para a resolução de uma determinada tarefa, pode com o passar do tempo, tornar-se ultrapassada e ineficiente. Perante a isso, com vistas a resolver os problemas e as necessidades em questão, esses esquemas podem ser mantidos em funcionamento, integrados a outros esquemas, ou substituídos através do planejamento de um processo de migração.

Este trabalho teve como proposta a contribuição com uma análise a cerca de ferramentas ETL disponíveis, somado ao planejamento e implementação de um estudo de caso com o objetivo de resolver os anseios de um modelo que apresenta problemas quanto ao seu funcionamento e utilização. Com isso, a adoção de um processo de migração baseado na modelagem de um novo esquema de armazenamento, associado a uma ferramenta ETL (como a PDI) acabaram por trazer a redução de tempo e complexidade na execução do projeto de migração de contextos de armazenamento, além de atender de maneira satisfatória as necessidades da empresa detentora dos dados.

No entanto, vale ressaltar que com relação a temas como integração de dados e migração de contextos de armazenamento, é notável a variedade de soluções existentes de acordo com o nível de complexidade e particularidade de cada projeto. Onde o problema descrito no estudo de caso, por não ser dotado de um nível alto de complexidade, poderia ser resolvido de outras formas. No entanto, as alternativas adotadas podem ser utilizadas como referência para a elaboração de estratégias em contextos semelhantes.

Portanto, isso abre precedentes para estudos futuros relacionados à pesquisas referentes à outras ferramentas e práticas disponíveis para a arquitetura de projetos que envolvam necessidades relacionadas à integração síncrona e assíncrona, podendo ser utilizadas na migração de contextos de armazenamento, ou integração contínua. Além disso, poderiam ser estudadas a implementação do estudo de caso com outras ferramentas além da PDI, a manutenção do ambiente ao longo do tempo, e as formas de utilização do produto gerado após a integração ou migração de esquemas de contextos de armazenamento.

6 REFERÊNCIAS

CASTANEDA R., FERLIN C., GOLDSCHIDTH R.,SOARES J. A., CARVALHO L. A. V., CHOREN R. Aprimorando Processo de Imputação Multivariada de Dados com Workflows. XXIII Simpósio Brasileiro de Banco de Dados – SBBD. 2008, São Paulo, Brasil.

CASTERS M., BOUMAN R., DONGEN J. V.,*Pentaho Kettle Solutions – Building Open Source ETL Solutions with Pentaho Data Integration* 1ª ed. Indianápolis: Wiley, 2010.

DATE C. J. *Introdução a Sistemas de Banco de Dados*. 8ª ed. Rio de Janeiro: Elsevier, 2004.

DE SORDI, J. O; MARINHO, B, L. Integração entre Sistemas: Análise das Abordagens Praticadas pelas Corporações Brasileiras. *Revista Brasileira de Gestão de Negócios*. São Paulo, v.9, n 23, p. 78-93, 2007.

FERREIRA J., MIRANDA M., ABELHA A., MACHADO J. O Processo ETL em Sistemas Data Warehouse. II Simpósio de Informática. 2010 Setembro: 757 -765. Braga, Portugal.

GARTNER. *Going beyond IT ROI – Estimating the business value of process integration solutions*. Irlanda: Gartner Group, fev. 2003.

GOOGLE – *Google Docs*. Disponível em: <http://www.google.com/apps/intl/pt-BR/business/docs.html>. Acesso em: 03 abr. 2012.

IBM, *IBM to Acquire Cognos to Accelerate Information on Demand Business Initiative*. Disponível em: <http://www-03.ibm.com/press/us/en/pressrelease/22572.wss>. Acesso em: 31 maio 2012.

_____., *Centro de Informações do IBM Cognos Business Intelligence 10.1.1*. Disponível em:<http://publib.boulder.ibm.com/infocenter/cbi/v10r1m1/index.jsp?topic=%2Fcom.ibm.swg.ba.cognos.ds_inst.10.1.1.doc%2Fc_ds_inst_installingdatamanagercomponentsononecomputer.html> . Acesso em: 11Abr. 2012.

_____., *Drivers de conexão*. Dísponível em: <http://publib.boulder.ibm.com/infocenter/rentrpt/v1r0m1/index.jsp?topic=%2Fcom.ibm.swg.ba.cognos.ug_ds.10.1.1.doc%2Fc_dbmsdrivers.html>. Acesso em: 31 maio 2012.

INFORMATICA – *Informatica PowerCenter 9.1.0 Getting Started (English)* – Disponível em: <http://mysupport.informatica.com.>. Acesso em: 12 abr. 2012

INMON W. H. *Building the Data Warehouse.* 3ª ed. Canadá: John Wiley & Sons, Inc. 2002.

JOHN P. J *Oracle Golden Gate 11g Implementer's guide* 1ª ed. ReinoUnido: Packt Publishing Ltd, 2011.

KIMBALL R. e CASERTA J. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data.* 1ª ed. EUA: Wiley Publishing, Inc. 2004.

LACY, Miguel Koren O'Brien. O Talento do Talend. Linux Magazine.São Paulo, a. 17, n.62, jan. 2009. Disponível em: < http://www.linuxnewmedia.com.br/images/uploads/pdf_aberto/LM_50_62_67_06_anali_talend.pdf>. Acesso em: 12 jun. 2012.

MICROSOFT – *Microsoft SQL Server Integration Services (SSIS) Connection.* Disponível em: <http://msdn.microsoft.com/en-us/library/ms140203.aspx>. Acesso em: 12 abr. 2012

MIQUEL L. H. *Oracle Fusion Middleware Getting Started with Oracle Data Integrator, Release 11g* (11.1.1), E12641-0, Setembro de 2010, Disponível em: <http://www.oracle.com/technetwork/middleware/data-integrator/overview/odigs-11g-168072.pdf>. Acesso em: 23 nov. 2011.

MIQUEL L. H. *Developer's Guide for Oracle Data Integrator, Release 11g* (11.1.1), E12643-03, Outubro de 2010. Disponível em: <http://docs.oracle.com/cd/E14571_01/integrate.1111/e12643.pdf>. Acesso em: 23 nov. 2011.

NANDA A. *Hands-On Microsoft SQL Server 2008 Integration Services* 1ª ed . EUA: The McGraw-Hill Companies. 2011

ORACLE, *Oracle Buys Golden Gate Software*. Disponível em: <http://www.oracle.com/us/corporate/press/022092>. Acesso em: 10 jun. 2012.

_____., *Matriz de drivers de conexão do GoldenGate*. Disponível em: <http://www.oracle.com/technetwork/middleware/data-integration/goldengate1111-cert-matrix-349179.xls>. Acesso em: 14 abr. 2012.

_____., *Oracle Buys Sunopsis*. Disponível em: <http://www.oracle.com/us/corporate/press/016802_EN>. Acesso em: 08 fev. 2012.

_____., *Matriz de drivers de conexão do ODI*. Disponível em: <http://www.oracle.com/technetwork/middleware/data-integrator/odi-11gr1certmatrix-163773.xls>. Acesso em: 14 abr. 2012.

PENTAHO - *Pentaho Conexões* – Disponível em: <http://wiki.pentaho.com/display/EAI/.03+Database+Connection>. Acesso em: 12 abr. 2012.

_____., *Pentaho Licenciamento* – Disponível em: <http://www.pentaho.com/license/> Acesso em: 12 abr. 2012.

_____., *Pentaho Community* - Disponível em: <http://community.pentaho.com/>. Acesso em: 12 abr. 2012.

_____., *Pentaho Kettle Project* - Disponível em: <http://kettle.pentaho.com/>. Acesso em: 12 abr. 2012.

_____., *Pentaho Fórum* – Disponível em: <http://forums.pentaho.com/>. Acesso em: 12 abr. 2012.

_____., *Pentaho Código Fonte* – Disponível em: <http://www.talendforge.org/trac/tos/>. Acesso em: 12 abr. 2012.

RAMEZ E. e NAVATHE S. B. *Sistemas de Banco de Dados*. 4ª ed. São Paulo: Pearson, 2005.

ROLDÁN M. C. *Pentaho 3.2 Data integration Begginer's Guide – Explore, transform, validate, and integrate your data withease.* 1ª ed. Bermingham: Packt Publishing, 2010.

SAP - *SAP compra Business Objects por US$ 6,8 bi* – Disponível em: < http://info.abril.com.br/aberto/infonews/102007/08102007-0.shl>. Acesso em: 03 mai. 2012.

_____., *SAP BusinessObjects Data Integrator Getting Started Guide*. Disponível em: < http://help.sap.com/businessobject/product_guides/xir2acc/en/DIGettingStartedGuide.pdf> Acesso em: 03 mai. 2012.

_____., *SAP BusinessObjects Data Integrator Core Tutorial*. Disponível em:<http://help.sap.com/businessobject/product_guides/xir2acc/en/dicoretutorial.pdf> Acesso em: 03 mai. 2012.

TALEND. *Talend Open Studio For Data Integration: User Guide*. Disponível em: <http://www.talend.com/products/open-studio-di.php> . Acessado em: 31 maio 2012.

TAURION, C. *Cloud Computing - Computação Em Nuvem - Transformando o Mundo da Tecnologia da Informação*. 1ª ed. Rio de Janeiro: Brasport, 2009.

WEKA - *Weka Project* . Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em: 23 maio 2012.

ANEXOS E APÊNDICES

Item 1: Script SQL de criação da base de dados destino conforme o MER da Figura 14:

```
CREATE TABLE aplicacoes
  (
     codigo    SERIAL NOT NULL,
     nome      CHARACTER varying(25) NOT NULL,
     descricao CHARACTER varying(500),
     PRIMARY KEY (codigo)
  );

CREATE TABLE filas_internas
  (
     codigo    SERIAL NOT NULL,
     nome      CHARACTER varying(25) NOT NULL,
     descricao CHARACTER varying(500),
     PRIMARY KEY (codigo)
  );

CREATE TABLE funcionalidades
  (
     codigo           SERIAL NOT NULL,
     codigo_aplicacao INTEGER,
     nome             CHARACTER varying(25) NOT NULL,
     descricao        CHARACTER varying(500),
     PRIMARY KEY (codigo),
     FOREIGN KEY (codigo_aplicacao) REFERENCES aplicacoes (codigo)
  );

CREATE TABLE status
  (
     codigo    SERIAL NOT NULL,
     nome      CHARACTER varying(25) NOT NULL,
     descricao CHARACTER varying(500),
     PRIMARY KEY (codigo)
  );

CREATE TABLE funcionarios
  (
```

```
    codigo SERIAL NOT NULL,
    nome   CHARACTER varying(50) NOT NULL,
    PRIMARY KEY (codigo)
  );

CREATE TABLE chamados
  (
    codigo                SERIAL NOT NULL,
    numero                INTEGER,
    codigo_status         INTEGER,
    codigo_aplicacao      INTEGER,
    codigo_funcionalidade INTEGER,
    codigo_funcionario    INTEGER,
    data_ini              DATE,
    data_fim              DATE,
    observacoes           CHARACTER varying(500),
    PRIMARY KEY (codigo),
    FOREIGN KEY (codigo_status) REFERENCES status (codigo),
    FOREIGN KEY (codigo_aplicacao) REFERENCES aplicacoes (codigo),
    FOREIGN KEY (codigo_funcionalidade) REFERENCES funcionalidades (codigo
),
    FOREIGN KEY (codigo_funcionario) REFERENCES funcionarios (codigo)
  );
```