MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

JULIO CESAR CARDOSO TESOLIN

A CARACTERIZAÇÃO DAS FONTES DE DADOS NA ESCOLHA DE ABORDAGENS DE INTEGRAÇÃO EM AMBIENTES BIG DATA

Rio de Janeiro
2016

INSTITUTO MILITAR DE ENGENHARIA

JULIO CESAR CARDOSO TESOLIN

A CARACTERIZAÇÃO DAS FONTES DE DADOS NA ESCOLHA DE ABORDAGENS DE INTEGRAÇÃO EM AMBIENTES BIG DATA

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientadora: Prof[a]. Maria Claudia R. Cavalcanti - D.Sc.

Rio de Janeiro
2016

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 - Praia Vermelha
Rio de Janeiro - RJ CEP 22290-270

005.1 Tesolin, Julio Cesar Cardoso
T337c A Caracterização das Fontes de Dados na Escolha de Abordagens de Integração em Ambientes Big Data / Julio Cesar Cardoso Tesolin, orientado por Maria Claudia R. Cavalcanti - Rio de Janeiro: Instituto Militar de Engenharia, 2016.

150p.: il.

Dissertação (mestrado) - Instituto Militar de Engenharia, Rio de Janeiro, 2016.

1. Curso de Sistemas e Computação - teses e dissertações. 1. Integração de Dados. I. Cavalcanti, Maria Claudia R. . II. Título. III. Instituto Militar de Engenharia.

INSTITUTO MILITAR DE ENGENHARIA

JULIO CESAR CARDOSO TESOLIN

A CARACTERIZAÇÃO DAS FONTES DE DADOS NA ESCOLHA DE ABORDAGENS DE INTEGRAÇÃO EM AMBIENTES BIG DATA

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientadora: Prof[a]. Maria Claudia R. Cavalcanti - D.Sc.

Aprovada em 28 de Novembro de 2016 pela seguinte Banca Examinadora:

Prof[a]. Maria Claudia R. Cavalcanti - D.Sc. do IME - Presidente

---

Prof. Ricardo Choren Noya - D.Sc. do IME

Prof[a]. Fernanda Araujo Baião Amorim - D.Sc. da UNIRIO

Prof[a]. Ana Carolina Salgado - Docteur da UFPE

Rio de Janeiro
2016

Aos meus familiares e amigos.

AGRADECIMENTOS

Nenhum trabalho de pesquisa é um esforço solitário. O mestrado não é diferente e esta peça não é só o resultado do esforço de seu autor, mas também da participação de familiares, amigos e professores.

Acredito que o início de tudo está em nossa base familiar. Agradeço aos meus pais, Elenilde e Antonio, por tudo que me ensinaram e ensinam até hoje. Seu constante encorajamento sempre foi decisivo. Espero algum dia chegar aos pés de sua integridade, resiliência, coragem e amor. Agradeço também à minha irmã Flávia e ao meu cunhado Fábio que me apoiaram desde o início desta empreitada e, espero, algum dia, ser uma inspiração para meu querido sobrinho João. Finalmente, agradeço à minha esposa Natalie pela sua compreensão e encorajamento durante este período. Sem sua ajuda, não teria logrado êxito.

Agradeço ao Instituto Militar de Engenharia pela oportunidade de retornar para mais um desafio. Voltar, ser bem recebido por seus integrantes e reencontrar velhos amigos de graduação é como retornar à própria casa após muito tempo longe. Ainda, agradeço à Seção de Sistemas e Computação da instituição, seus professores e colaboradores pela constante ajuda durante esta passagem.

Um agradecimento especial é devido à minha orientadora Maria Claudia Cavalcanti, carinhosamente conhecida como Yoko. Sem suas orientações e sua alegria na descoberta de novas fronteiras na área de gerenciamento de dados, certamente o trabalho não teria sido bem sucedido.

“É muito melhor arriscar coisas grandiosas, alcançar triunfos e glórias, mesmo expondo-se a derrotas, do que formar fila com os pobres de espírito, que nem sofrem muito, nem gozam muito, pois vivem nessa penumbra cinzenta que não conhece vitória nem derrota. ”

THEODORE ROOSEVELT

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

TPC-DI	Transaction Processing Performance Council Benchmark - Data Integration
UML	Unified Model Language

RESUMO

A era do *Big Data* é a consequência inevitável de nossa capacidade de gerar e coletar dados. Por consequência, as fontes de dados apresentam um comportamento mais dinâmico. Neste novo ambiente, o desafio de gerenciar o processo de integração de dados e o tráfego de rede entre as fontes de dados e os sistemas consumidores foi exacerbado pela quantidade de fontes e pelo volume de seu conteúdo, pela variedade de suas estruturas e de seus formatos e pela velocidade de surgimento de novas fontes para consumo.

Para enfrentá-los, uma sistemática de escolha e adaptação de abordagens de integração é desenvolvida, utilizando para isso a caracterização das fontes de dados, com o objetivo de minimizar a intervenção humana no processo de integração e reduzir o tráfego nas redes de comunicação que conectam os entes integrados. Esta sistemática é comparada com uma abordagem de materialização das fontes de dados no sistema de integração, utilizando um procedimento inspirado pelo TPC-DI como base para medição.

ABSTRACT

The Big Data era is an inevitable consequence of our capacity to generate and collect data. Because of that, data sources got a more dynamic behavior. In this new environment, the challenge to manage the data source's integration process and the network traffic between data producers and consumers has been overwhelmed by the number of data sources and their content's volume, by the variety of their structures and formats and by the velocity of their appearance.

To face these issues, this work presents a new method to help the users choosing data integration's approaches based on data sources characteristics. The goal is to reduce both human intervention in the integration process and the network traffic between the integrated peers. In order to evaluate the method, an integration environment was implemented based on TPC-DI sources and procedures.

1 INTRODUÇÃO

Geralmente, toma-se como certa a existência de dados nas mais diversas plataformas computacionais, porém pouco se questiona a sua procedência e como foram depositados nos elementos de persistência. Muitos sistemas precisam ser populados não só por dados provenientes da entidade a que serve, mas também de dados de outros domínios e organizações.

Contudo, a padronização na troca de dados entre sistemas não é obrigatória, sendo muitos deles desenvolvidos sem esta funcionalidade. É neste contexto em que se insere a área de integração de dados. Segundo Doan et al. (2012), a integração de dados é o processo de coletar, carregar e transformar dados de outras fontes, internas ou externas a uma organização, de tal sorte a enriquecer um determinado sistema. Apesar de crucial para o enriquecimento de tais sistemas, esta tarefa não é trivial. Segundo os mesmos autores, uma boa caracterização das fontes de dados a serem integradas é fundamental para escolher apropriadamente as abordagens de integração, seja ela a materialização ou a virtualização.

Muitos estudos e discussões foram realizados sobre a integração de dados nas últimas décadas. Aparentemente, o assunto havia esgotado ou diminuído sua relevância como objeto de estudos da academia. Contudo, o cenário mudou: a atual capacidade de gerar e coletar dados conferiu um caráter mais dinâmico às fontes de dados em três importantes vetores: volume, variedade e velocidade. Este novo ambiente deu origem a um movimento que ficou conhecido como *Big Data*.

Dessa forma, novos problemas surgiram e vários deles já estão sendo enfrentados pela comunidade acadêmica (DONG; SRIVASTAVA, 2013). Contudo, uma lacuna foi identificada: como escolher a abordagem de integração mais adequada para cada fonte de dados neste novo contexto? Observa-se que uma escolha inadequada da abordagem de integração pode levar a um tráfego de dados desnecessário nas redes de comunicação. Além disso, a gestão do processo de integração utilizando somente a intervenção humana pode não ser a mais adequada para gerenciar um volume maior de fontes de dados com características tão dinâmicas.

1.1 MOTIVAÇÃO

O termo *Big Data* ainda é um conceito abstrato, apesar da importância destes ambientes ter sido reconhecida por um amplo espectro da sociedade, inclusive pela comunidade acadêmica (ABADI et al., 2014). Em geral, o termo sugere um conjunto de dados que não podem ser percebidos, coletados, gerenciados e processados pelas ferramentas de *software* e *hardware* tradicionais em um tempo adequado (CHEN et al., 2014). Um ambiente *Big Data* é, normalmente, caracterizado pelo alto volume (fontes de dados e conteúdo), pela alta heterogeneidade (estruturas e formatos) e pela alta velocidade de surgimento de novos dados a serem processados.

Este fenômeno não parece ser meramente transitório. O relatório Beckman (ABADI et al., 2014), o último de uma série de relatórios dedicados à avaliação do estado da arte na área de gerenciamento de dados e produzido por renomados pesquisadores da área, elege o ambiente *Big Data* como o atual desafio a ser enfrentado pela comunidade acadêmica. Segundo o mesmo relatório, a necessidade de soluções desta natureza surgiram devido à convergência de três fatores principais: a redução do custo na geração dos dados, assim como de seu processamento, e a democratização dos dados. O primeiro fator está relacionado ao barateamento dos dispositivos de armazenamento e ao o surgimento dos sensores, dos dispositivos inteligentes (*smart devices*), das redes sociais e da *internet* das coisas, que conecta casas, carros, eletrodomésticos e vários outros dispositivos. Já o segundo está ligado aos avanços realizados nos processadores de múltiplas CPUs, no armazenamento de dados em dispositivos *solid state*, na computação em nuvem e nos programas de código livre. Finalmente, o último fator está relacionado ao envolvimento de outros atores no processo de geração, processamento e consumo de dados. Os agentes tomadores de decisão, os cientistas, os jornalistas e vários outros que eram considerados até então apenas como usuários finais estão agora intimamente ligados a todo processo, não sendo algo circunscrito apenas aos administradores de banco de dados e aos desenvolvedores.

Ainda segundo o relatório, vários itens relacionados à pesquisa da solução *Big Data* foram levantados na reunião. Como resultado final deste levantamento, os pesquisadores elegeram cinco grandes desafios a serem enfrentados pela comunidade científica de gerenciamento de dados: a construção de infraestruturas escaláveis para ambientes de grande volume e alta velocidade, o manejo da diversidade em ambientes de gerenciamento de dados, o processamento fim a fim e o entendimento do valor obtido com os dados, os serviços em nuvem e os papéis dos indivíduos no ciclo de vida dos dados. Como colocado pelos autores do relatório, os três primeiros desafios lidam com os principais atributos que

caracterizam uma solução *Big Data*: volume, velocidade e variedade.

Segundo Dong e Srivastava (2013), a integração de dados em um ambiente complexo e dinâmico de gerenciamento de dados difere das abordagens tradicionais nas mesmas dimensões que caracterizam a solução *Big Data*. Para eles, a integração de dados neste contexto é caracterizada pelo:

> **Volume:** Não só cada fonte de dado contém um tamanho bem superior ao de épocas anteriores, mas o número de fontes aumentou drasticamente. Mesmo para um mesmo domínio, o número de fontes pode chegar facilmente à casa dos milhares, algo certamente maior que o número de fontes que os sistemas tradicionais lidam;
>
> **Velocidade:** Como consequência direta do ritmo no qual os dados são coletados e continuamente colocados à disposição, muitas das fontes tornaram-se bem dinâmicas;
>
> **Variedade:** Mesmo fontes de dados de um mesmo domínio podem ser extremamente heterogêneas, seja no nível de esquema, seja como eles descrevem a mesma entidade no mundo real, exibindo considerável diferença mesmo em entidades substancialmente similares;
>
> **Veracidade:** Fontes de dados possuem qualidades bem diferentes, mesmo que estejam em um domínio comum, exibindo significantes particularidades na cobertura, na acurácia e no ciclo de vida do dado.

Ainda segundo estes pesquisadores, a importância da integração em ambientes *Big Data* levou a uma profusa pesquisa nos últimos anos em tópicos como mapeamento de esquemas (*schema mapping*), ligação de registros (*record linkage*) e fusão de dados (*data fusion*). No fechamento do artigo, os autores manifestam a preocupação pela falta de pesquisas na área de arquitetura de integração de dados nestes ambientes.

1.1.1 CENÁRIO MOTIVADOR

É recorrente utilizar cenários de operação de um sistema de telefonia celular como exemplo de aplicação de soluções *BigData*. Um exemplo didático é o gerenciamento dos elementos da rede de acesso de um operadora de telefonia celular. O elemento principal de uma rede de acesso é o setor de uma estação rádio base (ERB), popularmente chamado de "antena". É por meio da antena que o usuário obtém acesso e serviços da rede de telefonia móvel. Cada estação possui pelo menos uma antena.

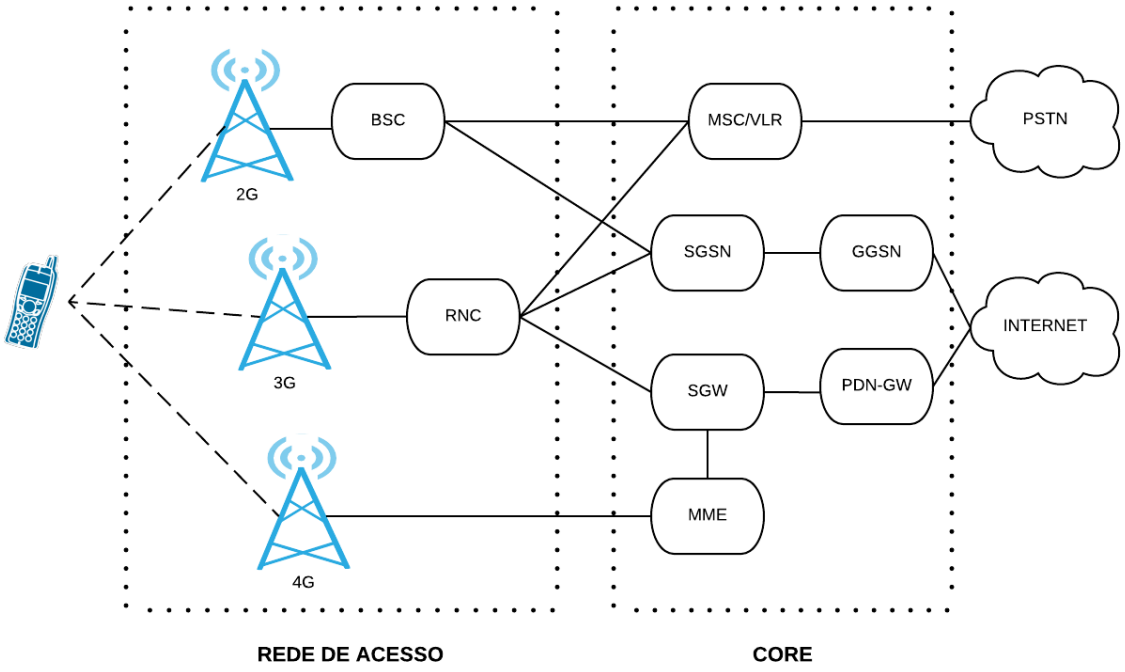

FIG. 1.1: Representação de uma Rede de Telefonia Celular

O setor é o elemento principal de medição no gerenciamento de uma rede de acesso. Cada setor possui configurações físicas e lógicas associadas. Além das configurações, cada setor possui medições de desempenho para avaliar a qualidade de cobertura e o serviço da operadora. Os setores são passíveis de falha, que precisam ser monitoradas. Sendo assim, cada setor gera informações de configuração, desempenho e falha. As configurações físicas, como altura da antena ou localização da estação, são informações que possuem baixa probabilidade de mudança. Já as configurações lógicas podem ser mudadas a qualquer momento, sendo necessário monitorá-las. Os dados gerados pelos setores para a verificação de desempenho e qualidade são os maiores volumes a serem trabalhados. Eles podem ser gerados periodicamente, podendo a cadência variar de dias para minutos e o seu volume pode variar de alguns KB até dezenas de TB. Da mesma forma que a configuração, os eventos de falha podem ocorrer a qualquer momento, sem prévio aviso.

Cada uma destas análises ocorre em uma determinada tecnologia: 2G, 3G e 4G. Como filosofia de um sistema de telefonia celular, uma chamada de voz e dados deve permanecer estabelecida a despeito da movimentação do usuário. Para que isso ocorra, os setores precisam ter conhecimento que são seus vizinhos de tal sorte a transferir a chamada para alguns deles quando o usuário se movimenta. Isso deve ocorrer tanto entre setores de uma mesma tecnologia como para qualquer outra. Logo, não só os relacionamentos

entre setores devem ser monitorados, como também o desempenho desta conexões (por exemplo, a quantidade de quebras de conexão).

O cenário brasileiro na telefonia celular coloca mais variáveis neste contexto. Devido ao número de operadoras e a extensão territorial, mais de um fabricante é utilizado para prover os equipamentos necessários para a construção de um sistema celular. Atualmente, exitem quatro grandes fabricantes que fornecem equipamentos para as operadora brasileiras. Apesar do protocolos de rede seguirem um padrão, a extração das informações de configuração, desempenho e falhas não precisa seguir de tal padronização. Além disso, à medida que novas tecnologias foram sendo criadas, novos formatos forma sendo criados. Enquanto os elementos da rede 2G não tinham compromisso com um formato de extração e integração, os equipamentos de tecnologia 3G e 4G já possuem extrações em formatos conhecidos como o XML e o JSON.

1.2 PROBLEMAS E HIPÓTESES

De acordo com o colocado no *caput* deste capítulo, há uma lacuna percebida na pesquisa sobre a área de integração de dados no novo contexto trazido pelos ambientes *Big Data* : como escolher a abordagem de integração apropriada? Esta indagação pode ser desmembrada em dois problemas principais que surgem quando a área de integração de dados defronta-se com ambientes tão complexos e dinâmicos :

Problema 1 : Uma escolha inadequada de abordagem de integração (materialização x virtualização) pode levar a um tráfego desnecessário nas redes de comunicação. É possível que a escolha e a alternância da abordagem de integração seja capaz de minimizar o tráfego de dados nas redes de comunicação?

Problema 2 : Uma vez que as fontes de dados tornaram-se mais numerosas e dinâmicas, não é adequado deixar a gestão do processo de integração de dados somente sob a supervisão humana. É possível criar um artefato computacional que altere certos aspectos do processo de integração sem a necessidade de intervenção humana?

Para estes dois problemas, a hipótese é que, a partir das características das fontes de dados, seja possível escolher dinamicamente a abordagem de integração mais adequada para cada uma delas ao longo do tempo de vida do ambiente de integração.

1.3 OBJETIVO

O objetivo deste trabalho é desenvolver uma sistemática de escolha e adaptação das abordagens de integração (materialização ou virtualização), utilizando para isso a caracterização das fontes de dados, com o intuito de minimizar a intervenção humana no processo de integração e reduzir o impacto do tráfego nas redes de comunicação que conectam os entes do ambiente de integração.

Especificamente, pretende:

- identificar as características das fontes de dados relevantes na escolha das abordagens de integração;
- criar uma sistematização que, a partir das características das fontes de dados, seja possível selecionar uma abordagem de integração adequada para cada uma delas;
- aplicar a sistematização em um estudo de caso de integração e confrontar os resultados com uma versão similar, porém utilizando apenas a materialização como abordagem de integração.

1.4 MÉTODO DE PESQUISA

Seguindo a orientação do trabalho de Wazlawick (2009), o método de pesquisa deste trabalho constitui-se em:

- Analisar a relevância das características das fontes de dados para a seleção de abordagens de integração. As características analisadas serão obtidas por meio do levantamento realizado na revisão bibliográfica e por meio de estudos empíricos, utilizando artefatos da UML como os diagramas de classe e de sequência;
- Construir uma classificação das características relevantes que possa representar, unicamente, as principais visões sobre o tema, conjugando aquelas observadas na revisão bibliográfica e os resultados obtidos nos estudos empíricos;
- Construir um fluxo de decisão baseado nas características relevantes das fontes de dados com o objetivo de selecionar a abordagem de integração mais adequada;
- Implementar uma solução de integração capaz de lidar dinamicamente com a materialização e a virtualização dos conteúdos das fontes de dados, utilizando para isso o fluxo de decisão construído;

– Testar a solução de integração utilizando como base o processo definido e os dados providos pelo *Transactional Process Council* em sua especificação de comparação de sistemas de integração (TPC-DI);

– Medir o volume de dados trafegados no ambiente de integração a cada ciclo de teste para determinar qual a configuração que minimizou a transferência de dados em rede virtual a ser construída;

1.5 ESTRUTURA

Este trabalho está organizado da seguinte forma:

Capítulo 2 : Descreve os tópicos da área de integração de dados pertinentes a este trabalho, como os processos, as arquiteturas e as abordagens, além do impacto resultante do movimento *Big Data*. Discorre ainda sobre a caracterização da fontes de dados, suas principais visões e a sua utilização no contexto da integração de dados;

Capítulo 3 : Neste capítulo, as características das fontes de dados levantadas na revisão bibliográfica são discutidas sob a ótica da escolha de uma abordagem de integração. Discorre ainda sobre aquelas não verificadas no levantamento e traz os estudos empíricos realizados para determinar se as mesmas seriam capazes de alterar a abordagem de integração. No final, apresenta as características relevantes para a seleção de abordagens e propõe um classificação para unificar as visões existentes;

Capítulo 4 : Descreve a proposta de uma solução de integração híbrida capaz de lidar tanto com a materialização quanto com a virtualização. Discorre ainda sobre cada módulo da solução e a implementação do fluxo de escolha de abordagens de integração;

Capítulo 5 : Discorre sobre as escolhas para a implementação da solução de integração e sobre o conjunto de fontes de dados a ser utilizado para verificação do objetivo. Coloca também as alterações realizadas nestas fontes para gerar uma maior diversidade de formatos. Descreve ainda a proposta de execução dos testes, suas fases e configurações, e, finalmente, apresenta o resultado de cada rodada realizada;

Capítulo 6 : Conclui o trabalho, destacando suas contribuições, limitações e propostas para trabalhos futuros.

2 INTEGRAÇÃO DE DADOS

De acordo com Doan et al. (2012), a integração de dados é um conjunto de técnicas que permitem a construção de sistemas flexíveis direcionados ao compartilhamento e à integração de múltiplos provedores de dados autônomos. Colocando de outra forma, o objetivo de um sistema de integração de dados é oferecer um acesso uniforme a um conjunto de dados autônomos e heterogêneos. Contudo, integrar dados não é uma tarefa trivial. Os mesmos autores mencionam três razões principais que dificultam a operacionalização de um sistema deste tipo: as sistêmicas, as lógicas e as sociais e administrativas. As dificuldades sistêmicas estão relacionadas ao desafio de habilitar diferentes sistemas para conversarem de forma transparente, enquanto as lógicas estão relacionadas a como os dados são logicamente organizados, mesmo dentro de um mesmo domínio. Já as dificuldades sociais e administrativas são ligadas ao acesso aos dados, devido a restrições de negócio, legais ou sociais. No contexto deste trabalho, apenas as questões sistêmicas e lógicas serão tratadas. A área de integração de dados é extensa em conteúdo e este capítulo concentra-se apenas em alguns aspectos das arquiteturas de ambientes de integração de dados, na caracterização das fontes de dados e na sua efetiva utilização na construção e na operação destes ambientes.

2.1 ARQUITETURA DE AMBIENTES DE INTEGRAÇÃO DE DADOS

Segundo Russom (2008), há a necessidade que arquiteturas suportem implementações de ambientes de integração de dados, uma vez que o volume, a velocidade e a variedade tornaram o ambiente por demais complexo, não podendo mais ser mais tratado como um apêndice de soluções como as de *data warehousing*. A revisão bibliográfica apresentou várias referências sobre padrões arquiteturais para solucionar os problemas de integração de dados. Contudo, não foi identificada uma estrutura formal para o desenvolvimento destas arquiteturas e que, por conseguinte, possa estabelecer um padrão na nomenclatura para as diversas soluções encontradas na literatura. Enquanto alguns autores chamam certas soluções de padrões arquiteturais (GIORDANO, 2011), outros as denominam apenas como técnicas e tecnologias de integração de dados (WHITE, 2006). Para evitar a utilização de denominações exclusivas de uma indústria e, assim, incorrer em erros ou

falhas de interpretação, este trabalho recorreu às definições básicas de estruturas arquiteturais, utilizando para tanto a pesquisa de *frameworks* de construção de arquitetura como o TOGAF (HAREN, 2011) e o ISO/IEC 42010:2007 (ISO/IEC/IEEE, 2011). Neste sentido, esta seção foi dividida em duas partes. A primeira apresenta os ambientes de integração tradicionais descritos na literatura, enquanto a segunda apresenta alguns aspectos construtivos a serem observados, utilizando o *framework* definido pelo TOGAF como referência.

2.1.1 PADRÕES ARQUITETURAIS TRADICIONAIS

Segundo Doan et al. (2012), há várias possibilidades de arquiteturas de integração de dados, sendo que a maioria repousa em um espectro entre a materialização e a virtualização. De um lado deste espectro, em uma abordagem de integração puramente **materializada**, os conteúdos das fontes de dados são carregados e internalizados em um repositório de dados físico, onde as consultas provenientes dos sistemas consumidores são respondidas. Do outro lado deste espectro, em uma abordagem de integração puramente **virtualizada**, os dados permanecem em sua fonte original até que seja necessário acessá-los. Apesar das diferenças entre as abordagens, muitos dos desafios encontrados na construção destes ambientes de integração são compartilhados.

Ainda segundo os mesmos autores, a Figura 2.1 representa um ambiente genérico de integração de dados, onde estão representados os componentes lógicos normalmente encontrados, independente da abordagem de integração sendo utilizada. Em sua parte inferior, a Figura 2.1 representa as **fontes de dados**. Estes são os repositórios onde os **conteúdos** de interesse repousam, possuindo diferenças em vários níveis, como o modelo lógico utilizado para representar o conteúdo ou a capacidade do repositório de responder a consultas de outros sistemas, entre outras características. Acima das fontes de dados estão os programas cujo papel é comunicar-se com as fontes de dados. Na virtualização, estes programas são chamados de ***wrappers*** e seu papel é enviar consultas à fonte de dados, receber as respostas e, possivelmente, aplicar transformações básicas em cima destas respostas. O usuário interage com a solução de integração por meio de um esquema único, que neste caso é denominado de esquema de mediação. Este esquema é construído para o ambiente de integração de dados e contem somente os aspectos de domínio que são relevantes para o próprio ambiente. Sendo assim, não é necessário que contenha todos os atributos vistos nas fontes, mas somente um subconjunto deles. Nota-se que na virtualização, o esquema de mediação não é criado para guardar qualquer dado. Sua função é

responder às consultas feitas pelos sistemas consumidores que o utilizam, representando apenas um conjunto lógico dos dados a serem integrados. Já na materialização, ao invés de utilizar um esquema de mediação, o usuário aplica consultas em termos de um esquema onde os conteúdos das fontes de dados são fisicamente internalizados. Ao invés de *wrappers*, estas soluções incluem módulos de processamento mais complexos denominadas ETL (*Extract-Transform-Load*) ou ***extractors***, que periodicamente extraem dados das fontes e carregam em um esquema físico. Diferente daqueles, os *extractors* aplicam, tipicamente, transformações mais complexas para os dados, que pode envolver limpeza, agregação e mudança de valores. Estas transformações fazem o papel do mapeamento de esquemas na abordagem de virtualização, mas tendem a ter um comportamento mais procedimental por natureza.

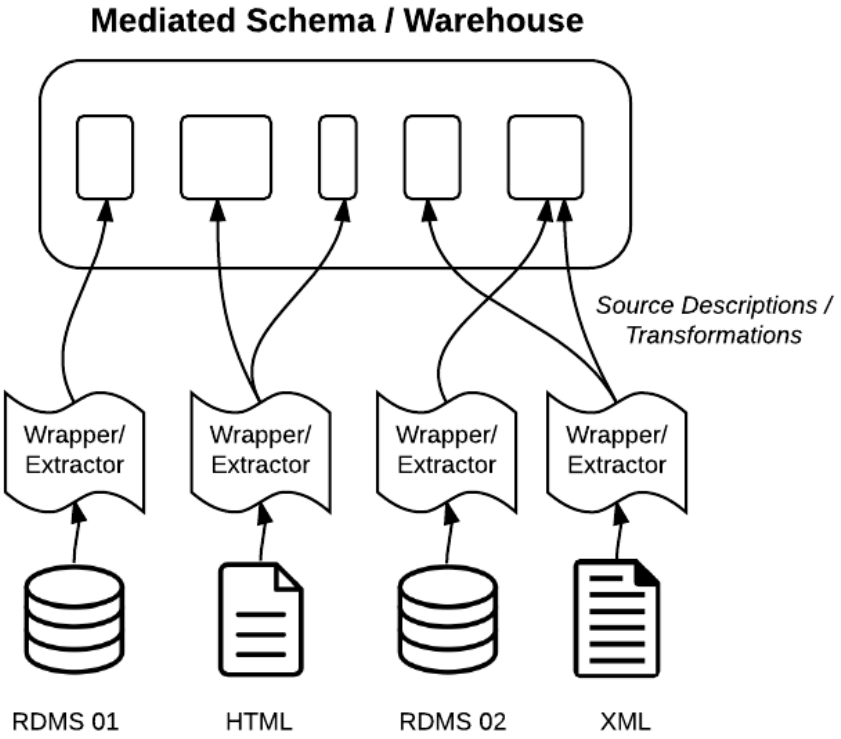

FIG. 2.1: Ambiente Genérico de Integração segundo Doan et al. (2012)

Para Doan et al. (2012), a chave para construir ambientes de integração de dados é a descrição de suas fontes de dados. Estas descrições especificam as propriedades das fontes que o sistema precisa conhecer para utilizar seu conteúdo. O componente principal das descrições das fontes é o seu mapeamento semântico, que relaciona os esquemas das

fontes de dados com o esquema de mediação. O mapeamento semântico especifica como os atributos nas fontes correspondem aos atributos do esquema de mediação e como os diferentes agrupamentos em tabelas são resolvidos.

Comparada à virtualização, a materialização permite a execução de consultas computacionalmente mais intensivas sobre grandes volumes de dados, especialmente em dados históricos. Por outro lado, a materialização impõe um atraso entre a atualização do conteúdo da fonte de dados e a efetiva disponibilização para os sistemas consumidores. O que se observa na prática (DOAN et al., 2012) é que *existem várias abordagens híbridas para resolver estes problemas*. Invariavelmente, algumas fontes e relacionamentos são atualizados mais frequentemente do que outros. Logo, a maioria das soluções de integração explora a materialização parcial e o aprovisionamento prévio de resultados que não precisam estar completamente atualizados ou que dificilmente se alteram ao longo do tempo, usando-se a virtualização apenas para fontes mais dinâmicas.

Trabalhos recentes (KHAZANKIN; DUSTDAR, 2010)(MEISEN et al., 2013) referenciam o artigo de White (2006), onde o autor sugere uma classificação que torna mais claro o comportamento da fontes de dados e as correlaciona com as abordagens existentes, descrevendo as seguintes técnicas: consolidação, federação e propagação. Na técnica de consolidação, dados de múltiplas fontes são capturados e integrados em um único repositório de dados. Na federação, uma única visão virtual é criada a partir de uma ou mais fontes, onde a recuperação dos dados é sempre feita por demanda. Finalmente, na técnica de propagação, a captura dos dados é ativada por eventos específicos na fonte, enviando os dados para os sistemas consumidores. Ainda no mesmo artigo, o autor lista as principais tecnologias de implementação destas técnicas. São denominadas *Extract, Transform and Load* (ETL), *Enterprise Information Integration* (EII) e *Enterprise Architecture Iniciative* (EAI). A tecnologia ETL extrai dados dos sistemas-fonte, transformando-os e carregando-os no repositório alvo. As fontes e os alvos geralmente são bancos de dados e arquivos, porém esta tecnologia é capaz de lidar com outros tipos de repositórios de dados, como por exemplo, filas de mensagens. Já a tecnologia EII provê uma visão virtual de vários dados dispersos, sendo usada por consultas geradas por demanda para acessar dados operacionais, *data warehouses* ou informações não estruturadas. Finalmente, a tecnologia EAI integra sistemas, permitindo a comunicação e a troca de transações, mensagens e dados, com cada um usando interfaces padronizadas. Ela permite o acesso ao dado de forma transparente, sem saber sua localização ou formato. Nota-se pelas descrições que as técnicas de propagação e consolidação são destinadas à materialização do conteúdo das fontes de dados, enquanto a técnica de federação implementa a abordagem de virtualização.

Esta visão é compartilhada por Giordano (2011), porém o autor as denomina de padrões de arquiteturais. Adicionalmente, o autor propõe uma arquitetura de referência para a construção de ambientes de integração de dados, como mostra a Figura 2.2.

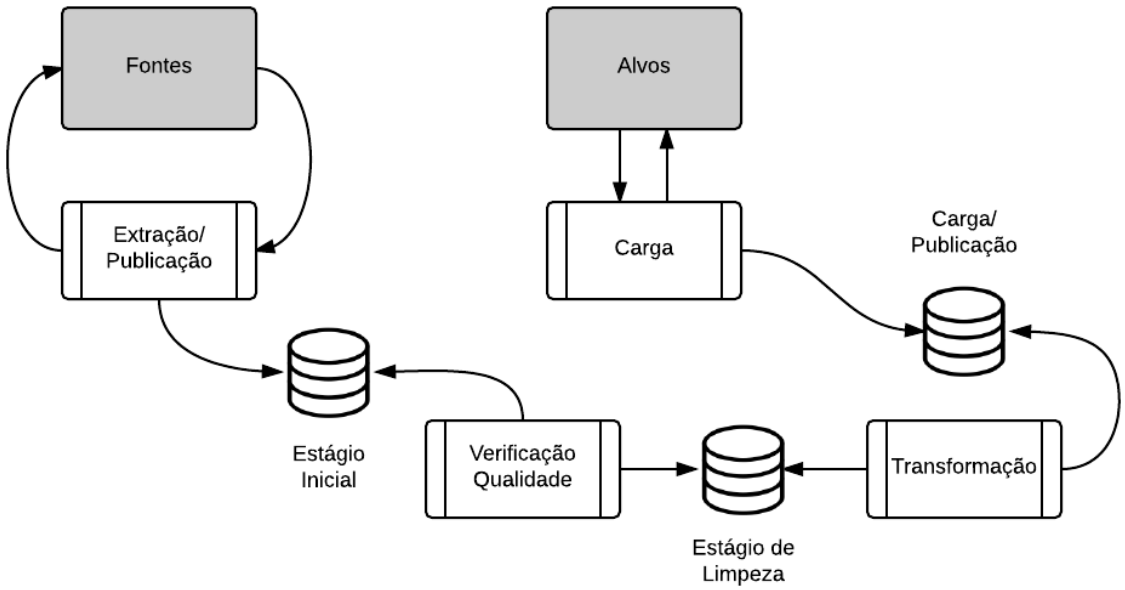

FIG. 2.2: Arquitetura de Referência para Ambientes de Integração de Dados (GIORDANO, 2011)

A arquitetura de referência mostrada na Figura 2.2, define os processos e os repositórios que suportam a captura, a verificação da qualidade, o processamento e a movimentação dos dados, sejam eles transacionais ou em lote, para um ou mais sistemas alvo. Segundo o autor, há dois objetivos a serem alcançados por esta arquitetura de referência: simplicidade e escalabilidade.

De acordo com o autor, comunicar conceitos comumente aceitos é fator chave para o sucesso de qualquer projeto, seja ele de integração de dados ou um projeto de um banco de dados relacional. Uma parte do sucesso do modelagem de dados com diagramas entidade-relacionamento é a simplicidade da notação e do seu entendimento. Da mesma forma, as camadas comuns da arquitetura de integração são desenvolvidas para prover o mesmo meio de comunicação e entendimento sobre os estágios e os processos encontrados no desenvolvimento. Usando a arquitetura de referência, sempre haverá uma camada de extração suportada por um estágio (ou repositório) inicial, que será fonte para a camada de verificação de qualidade. Os dados verificados ficam disponíveis em um repositório apropriado, que será fonte para a camada de transformação. Após a transformação,

os dados repousam em um repositório de carga e publicação que, finalmente, levará a uma camada responsável pela carga no sistema alvo. Cada camada ou estágio tem um propósito e um uso específico bem definido, o que reforça o conceito de reusabilidade. É importante salientar que cada uma destas camadas não é necessariamente sequencial ou até mesmo necessária. Nem todo processo de integração exigirá transformações ou até mesmo verificação de qualidade, dependendo apenas dos requisitos de negócio daquele processo de integração em particular.

Já os requisitos para escalabilidade e estabilidade têm crescido consideravelmente nos últimos anos. Ambientes de inteligência de negócios como *data warehouses* corporativos, que demandam ambientes analíticos globais, certamente não podem ficar fora por semanas ou dias. Eles precisam estar disponíveis para um grupo muito maior de pessoas que precisam diariamente acessá-los para desenvolver seus trabalhos. Para lidar com este crescimento de dados e a necessidade de períodos de interrupção (*outage*) cada vez mais curtos, a arquitetura de referência também foi desenvolvida como uma planta lógica que pode ser instanciada por meio de uma ou mais máquinas físicas, provendo a habilidade de limitar a escalabilidade para somente um número de CPUs agrupados, como mostra a Figura 2.3.

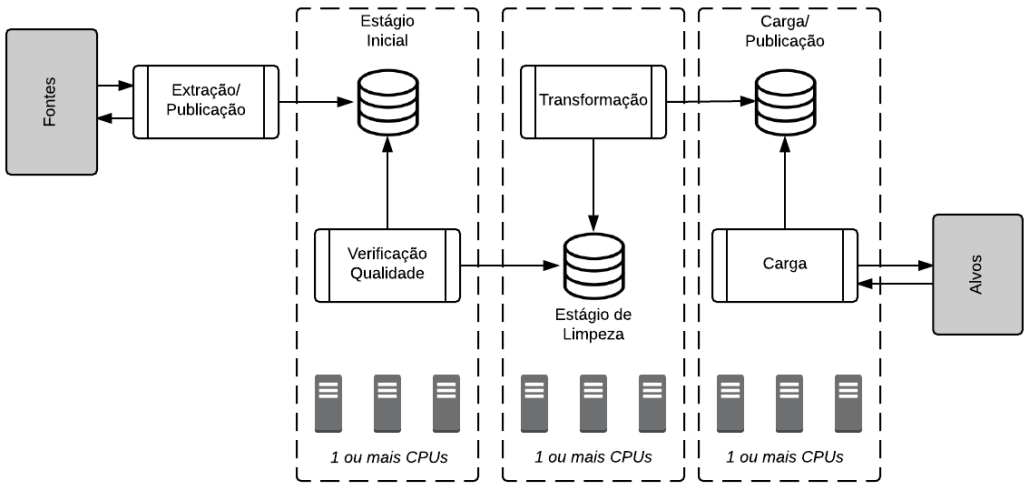

FIG. 2.3: Proposta de Escalabilidade a partir da Arquitetura de Referência (GIORDANO, 2011)

Como muitas definições e vocabulários importantes neste trabalho de pesquisa foram apresentados nesta seção, a lista abaixo apresenta um resumo de cada um deles e servirá como referência ao longo do restante deste documento.

– **Virtualização**: é a abordagem de integração onde os dados permanecem em seu

repositório original até que seja necessário acessá-los;

- **Materialização**: é a abordagem de integração onde os dados são carregados e internalizados em um repositório de dados físico, independente se há a necessidade de consumi-los no momento em que ocorre a extração;
- **Fonte de Dados**: é o repositório ou o sistema onde o conteúdo repousa;
- **Alvo**: é o sistema interessado pelo conteúdo da fonte de dados. Também é denominado por *Sistema Consumidor*;
- **Conteúdo**: são os dados propriamente ditos. Também denominado por *carga útil*;
- ***Wrapper* ou Tradutor**: é o programa responsável por conectar-se a fonte de dados e entregá-los ao sistema consumidor após solicitação;
- ***Extractor* ou Extrator**: é o programa responsável pela extração periódica de dados (conteúdos) das fontes e pela internalização em um ou vários repositórios físicos, permitindo o acesso ao sistema consumidor;

2.1.2 ARQUITETURA HÍBRIDA DE INTEGRAÇÃO DE DADOS

No contexto de integração de várias fontes de dados, como em um ambiente *Big Data*, a utilização de uma única abordagem de integração pode não ser a estratégia mais adequada. Como colocado na seção anterior, as fontes podem possuir comportamentos e características tais que os sistemas consumidores e a própria solução de integração se beneficiariam caso um parcela fosse integrada por meio de uma abordagem de virtualização, enquanto a outra utilizasse a de materialização.

Esta visão corrobora as colocações de Hull e Zhou (1996) no seu artigo *A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches*, uma das referências no estudo de arquitetura híbridas de integração de dados ou de materialização parcial. Nele, os autores sugerem a utilização da materialização parcial nos casos onde as fontes de dados não possuem uma atualização frequente de seu conteúdo e existe a necessidade de responder rapidamente às consultas feitas pelos sistemas consumidores. Outros artigos como (ASHISH et al., 1999) seguem esta visão e sugerem a implementação de arquiteturas híbridas dedicadas a solucionar um problema específico. A Figura 2.4 representa, genericamente, a arquitetura híbrida de integração proposta pelos autores anteriormente citados.

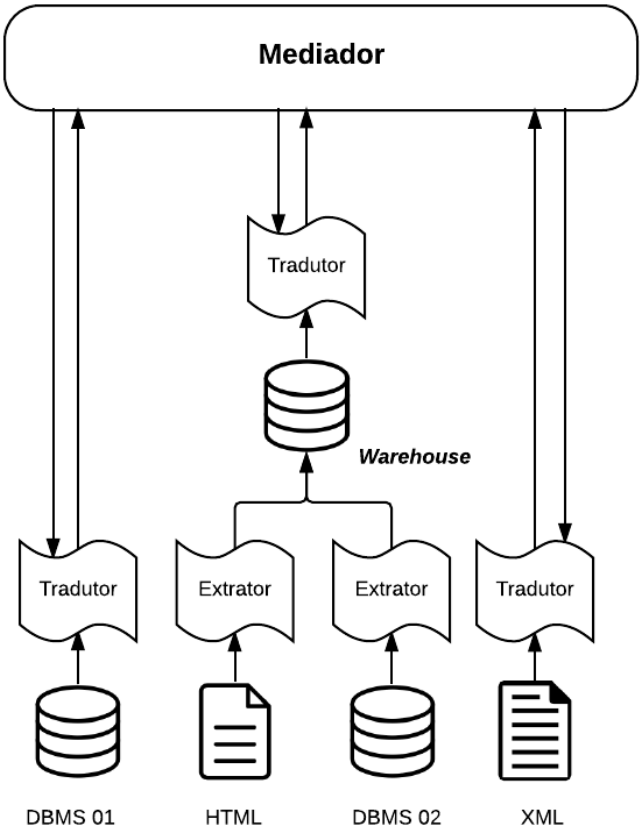

FIG. 2.4: Arquitetura Híbrida de Integração de Dados

2.1.3 ASPECTOS DA ESTRUTURA ARQUITETURAL PARA AMBIENTES DE INTEGRAÇÃO DE DADOS

Após o levantamento das estruturas de desenvolvimento de arquiteturas existentes, o *The Open Group Architecture Framework* (TOGAF) foi escolhido como referência para a discussão de certos aspectos relevantes para a área de integração de dados. A organização gestora do TOGAF define uma estrutura arquitetural como aquela que provê uma fundação que pode ser usada no desenvolvimento de uma gama de diferentes arquiteturas. Ela deve ser capaz de descrever um método para projetar um determinado estado de uma corporação, em termos de um conjunto de blocos construtivos, mostrando como os mesmos se encaixam. Devem também conter um conjunto de ferramentas e um vocabulário comum, além de uma lista de padrões recomendados e de produtos adequados que podem ser usados para implementar estes blocos construtivos.

Ainda segundo o TOGAF, existem quatro domínios de arquitetura que são comumente aceitos como subconjuntos de um arquitetura corporativa: o negocial, o de dados,

o de aplicações e o de tecnologia. A arquitetura negocial define as estratégias, a governança, a organização e os processos-chave da corporação. Já a arquitetura de dados descreve as estruturas lógicas e físicas dos ativos de dados da organização e seus recursos de gerenciamento, enquanto que a arquitetura de aplicações provê a planta base para que aplicações individuais possam ser implementadas, suas interações e seus relacionamentos com o núcleo do processo negocial da organização. Finalmente, a arquitetura de tecnologia descreve os programas e as capacidades computacionais necessárias para suportar a implementação dos serviços negociais, de dados e de suas aplicações. Isso inclui a infraestrutura de tecnologia de informação, de mediação, as redes de comunicação etc.

O estudo dos domínios de arquitetura deste *framework* foge ao escopo deste trabalho. Apenas alguns aspectos do domínio da arquitetura de aplicações, de dados e de tecnologia, relevantes para a arquitetura de uma solução de integração de dados, serão tratados nas próximas seções.

2.1.4 ASPECTOS DO DOMÍNIO DA ARQUITETURA DE APLICAÇÕES

De acordo com a versão 9.1 do TOGAF, um dos aspectos verificados no domínio da arquitetura de aplicações é o fluxo de dados entre os elementos participantes do sistema. Genericamente, os entes podem se comunicar de acordo com duas abordagens: ponto-a-ponto e *hub-and-spoke*. Na comunicação ponto-a-ponto, os entes se comunicam diretamente, enquanto que na comunicação *hub-and-spoke* existe um elemento de mediação entre os mesmos. Russom (2008) explora vigorosamente esta característica de comunicação em seu artigo. Para o autor, o conceito de *hub-and-spoke* é fácil de entender e trabalhar, podendo ser expressado por infinitas variações, o que reforça o seu reuso. E, em comparação com a comunicação ponto-a-ponto, o número de interfaces é drasticamente reduzido em casos limites. A Figura 2.5 mostra as duas situações em um ambiente de integração.

Na arquitetura representada no item (a) da Figura 2.5, as fontes de dados são conectadas diretamente aos sistemas consumidores. Para este caso, o número de conexões entre as fontes de dados e os sistemas consumidores pode ser dado pela seguinte formulação:

$$n_c = \binom{2}{n_s + n_t} - \binom{2}{n_s} - \binom{2}{n_t} = n_s * n_t$$

onde:

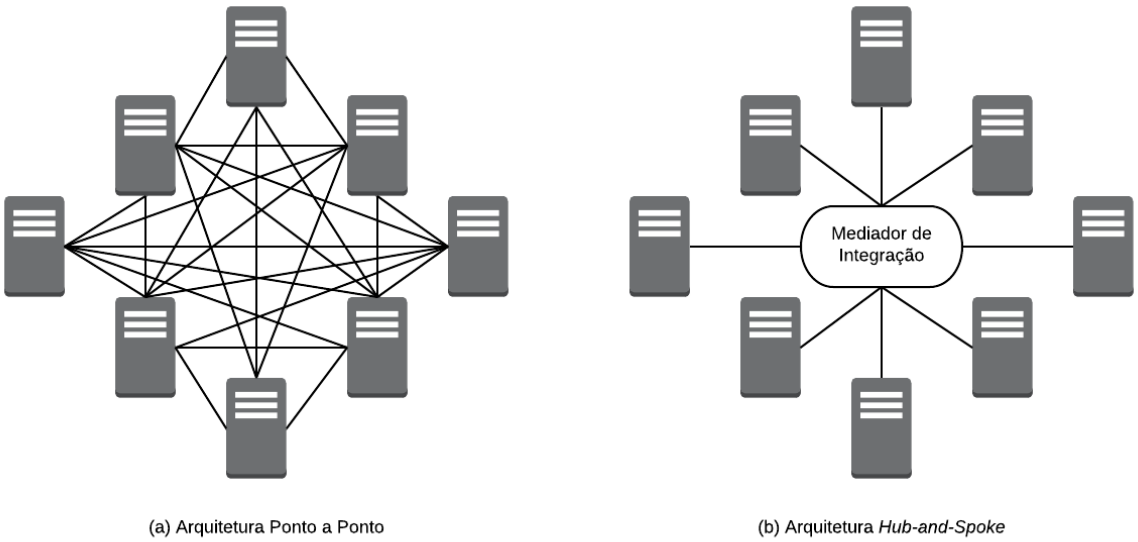

FIG. 2.5: Comunicação entre Servidores (RUSSOM, 2008)

$$n_c = \text{número de conexões},$$
$$n_t = \text{número de alvos},$$
$$n_s = \text{número de fontes}$$

Já a arquitetura representada no item (b) da Figura 2.5, as fontes de dados e os sistemas consumidores são conectados por um elemento de mediação. Desta forma, o número de conexões pode ser representado pela seguinte formulação:

$$n_c = \binom{1}{n_s} + \binom{1}{n_t} = n_s + n_t$$

Como se pode depreender das duas formulações, se houver apenas um sistema consumidor, o número de conexões entre este e as fontes de dados é praticamente o mesmo. Contudo, a medida que uma solução de integração sirva a mais e mais sistemas consumidores, o número de conexões em um arquitetura ponto-a-ponto aumenta drasticamente em relação ao esquema de *hub-and-spoke*. Apesar desta vantagem colocada por Russom, soluções que utilizam exclusivamente comunicações do tipo *hub-and-spoke* são difíceis de serem encontradas, uma vez que a introdução de um elemento de mediação aumenta a latência de recuperação do conteúdo das fontes de dados. Segundo o autor, as soluções encontradas para lidar com ambientes complexos e dinâmicos de gerenciamento de dados

possuem um abordagem híbrida, permitindo que algumas fontes possam se comunicar diretamente com os sistemas consumidores.

2.1.5 ASPECTOS DO DOMÍNIO DA ARQUITETURA DE DADOS

Como visto na Seção 2.1.1, o processo de integração pode utilizar repositórios temporários para guardar o conteúdo extraído e adequá-lo ao esquema dos sistemas consumidores. Para cada etapa do processo, um repositório diferente pode ser utilizado, sendo que estas estruturas de guarda podem variar desde simples coleções de arquivos brutos organizados em diretórios a sistemas gerenciadores de bancos de dados, dependendo da complexidade exigida pelo projeto de integração. Esta seção foca apenas nos repositórios inicial e final do processo de integração, pois são as estruturas de guarda a serem utilizadas neste trabalho.

Considerando que um SGBD é a opção mais adequada para a guarda dos conteúdos em comparação com a guarda dos mesmos em arquivos brutos, há de se questionar se um mesmo tipo é adequado para cada fase do processo. Como colocado por Sadalage e Fowler (2012), o tradicional sistema relacional possui a característica natural de integração, uma vez que o modelo é largamente usado e sua linguagem de consulta (SQL) é simples e padronizada. Dessa forma, qualquer sistema consumidor é capaz de internalizar o conteúdo já integrado. Contudo, a utilização de um SGBD na etapa de extração revela-se um desafio, uma vez que o atual contexto de geração de dados provê conteúdos descritos nos mais variados modelos e formatos, muitos deles não compatíveis com o modelo relacional.

Ainda como colocado por estes autores, a exigência crescente de troca de informações entre sistemas leva à necessidade de utilizar formas agregadas de representação dos conteúdos, uma vez que diminui a necessidade de buscas contínuas e repetidas para sua total extração. Porém, neste caso, a utilização de um SGBD relacional como um repositório temporário da extração do conteúdo das fontes de dados impõe o uso de uma das seguintes opções: transformar o modelo lógico corrente, geralmente agregado, para um relacional ou simplesmente internalizá-los como meros objetos binários (*BLOBs*).

A primeira opção pode parecer adequada, uma vez que o repositório final do processo de integração mais adequado para a troca de informações possui um modelo relacional. Contudo, questiona-se a validade desta transformação apenas para guardar o conteúdo extraído em um repositório como um SGBD relacional, uma vez que os ganhos conquistados ao colocar o conteúdo neste sistema, ao invés de um mero arquivo, podem ser eliminados pela complexidade de transformá-lo. A segunda opção elimina a transformação do modelo, guardando o conteúdo como um atributo do tipo objeto em uma tabela. Porém,

mais uma vez, as vantagens de colocá-lo em um sistema gerenciador de banco de dados ao invés de guardá-lo em arquivo podem ser reduzidas ou praticamente eliminadas. Isso decorre do fato que os atuais sistemas relacionais não conseguem, até o presente momento, reconhecer a sintaxe existente naquele campo.

Em Liu e Gawlick (2015), os autores advogam sobre a necessidade dos tradicionais bancos relacionais evoluírem de uma abordagem esquema primeiro-dados depois (*schema first-data later*) para uma abordagem dados primeiro-esquema depois (*data first-schema later/never*). Eles definem os atuais conteúdos sendo disponibilizados de **dados de esquema flexível** (*Flexible Schema Data*) e proveem os princípios e as práticas necessárias para lidar com tal cenário. No mesmo sentido, Sadalage e Fowler (2012) apresentam os bancos de dados NoSQL como um movimento tecnológico na área de gerenciamento de dados que possui como uma de suas características a inexistência de esquema explicitamente definido como nos bancos de dados relacionais. Os autores alertam para o fato do abuso de linguagem ao defini-los como bancos de dados sem esquema, uma vez que cada conteúdo possui um esquema implícito na sua descrição. Em Strnad et al. (2013), os autores mostram a utilização de um SGBD NoSQL do tipo chave valor para o gerenciamento de conteúdos descritos no formato XML.

2.1.6 ASPECTOS DO DOMÍNIO DA ARQUITETURA DE TECNOLOGIA

Um aspecto do domínio da arquitetura de tecnologia relevante para ambientes *Big Data* é a capacidade dos artefatos de *software* e *hardware* escalarem. Segundo (BONDI, 2000), escalabilidade é a capacidade de um sistema, rede ou processo tem de lidar com o crescimento do trabalho computacional ou o seu potencial de ampliação para acomodar tal crescimento. Geralmente, os métodos para adicionar recursos a uma aplicação caem em duas grandes categorias: o escalonamento horizontal e o escalonamento vertical Michael et al. (2007).

Escalar horizontalmente (ou *scale-out*) significa adicionar mais nós em um sistema, como, por exemplo, adicionar um novo computador a uma aplicação de processamento distribuído. Dessa forma, é possível que algumas centenas de pequenos computadores organizados em grupos (*clusters*) possa obter um poder computacional agregado superior aos de computadores de grande porte tradicionais. Esta capacidade só foi possível com o desenvolvimento de enlaces de rede de alto desempenho, aumentando a demanda por programas que permitam a manutenção e o gerenciamento eficiente de múltiplos nós assim como de repositórios compartilhados de dados que possuam alto desempenho nas

operações de entrada e saída. Já escalar verticalmente (ou *scale-up*) significa adicionar recursos a um único nó em um sistema, tipicamente envolvendo a adição de mais núcleos (CPUs), memória ou espaço em disco a um único computador. O escalonamento vertical destes sistemas também permite o uso mais eficiente das tecnologias de virtualização, uma vez que provê mais recursos para um mesmo conjunto hospedado de sistemas operacionais e aplicativos.

Há dois padrões arquiteturais antagônicos que ilustram as opções de escalonamento anteriores. São os de total compartilhamento (*Share-Everything*) e os sem compartilhamento (*Share-Nothing*). No padrão de total compartilhamento, um único computador resolve os problemas de complexidade de processamento usando os mesmos recursos de memória, espaço em disco e processadores, caraterísticas estas que demonstram um padrão arquitetural de escalonamento vertical. Este tipo de arranjo é conhecido por multiprocessamento simétrico (*Symmetric Multiprocessing*-SMP). Ao longo do tempo, algumas variantes desta arquitetura foram criadas, utilizando algum tipo de compartilhamento, seja de memória, seja de discos. Já o arranjo sem compartilhamento usa um conjunto relativamente independente de servidores para trabalharem cooperativamente em um subconjunto de um problema, onde, ocasionalmente, estes servidores precisarão compartilhar dados por meio de uma conexão de alta velocidade. Este sistemas ficaram famosos por sua capacidade de escalonamento vertical, que em suas versões mais avançadas tornaram-se conhecidos como arranjos de processamento paralelo massivo (*Massive Parallel Processing*-MPP).

Stonebraker e Cattell (2011) discutem a utilização destas arquiteturas no mundo dos sistemas gerenciadores de banco de dados. As primeiras versões destes sistemas usaram arranjos de escalonamento vertical puro para prover as soluções. Com o aumento da demanda por dados, variantes foram utilizadas, recorrendo assim ao compartilhamento de memória ou de discos para resolver os problemas derivados do volume. Em todos estes casos, existe um limitante inerente para escalá-los, seja por questões relacionadas ‘a latência de memória, seja por questões de natureza construtiva como, por exemplo, a sincronização de discos em um arranjo onde os mesmos são compartilhados. Para os autores, com a perspectiva de um aumento ainda maior da geração de dados e a necessidade de prover respostas rápidas a partir desta realidade, os arranjos de escalonamento vertical são os mais adequados para prover as soluções neste contexto de gerenciamento de dados.

2.2 A CARACTERIZAÇÃO DAS FONTES DE DADOS

Vários trabalhos recentes fornecem propostas de caracterização das fontes de dados, como pode se visto em de Faria Cordeiro (2015). Contudo, percebe-se que todos eles possuem uma referência em comum: o trabalho clássico de Sheth e Larson (1990) sobre federação de bancos de dados. Posteriormente, os conceitos foram revisitados, refinados e estendidos por outro artigo de Sheth (1999). Neste artigo, o autor discorre sobre a interoperabilidade de sistemas e os três vetores que os caracterizam: a distribuição, a autonomia e a heterogeneidade. Outro trabalho relevante no estudo da caracterização das fontes de dados, porém com uma abordagem distinta, pode ser visto na publicação de Hohpe e Woolf (2003) sobre a integração de sistemas. As subseções seguintes resumem os principais pontos destas duas visões.

2.2.1 INTEROPERABILIDADE DE SISTEMAS

Assim como em seu artigo clássico sobre federação de banco de dados (SHETH; LARSON, 1990), Sheth (SHETH, 1999) utiliza três vetores ortogonais para descrever e analisar a interoperabilidade de sistemas: a distribuição, a autonomia e a heterogeneidade (Figura 2.6)

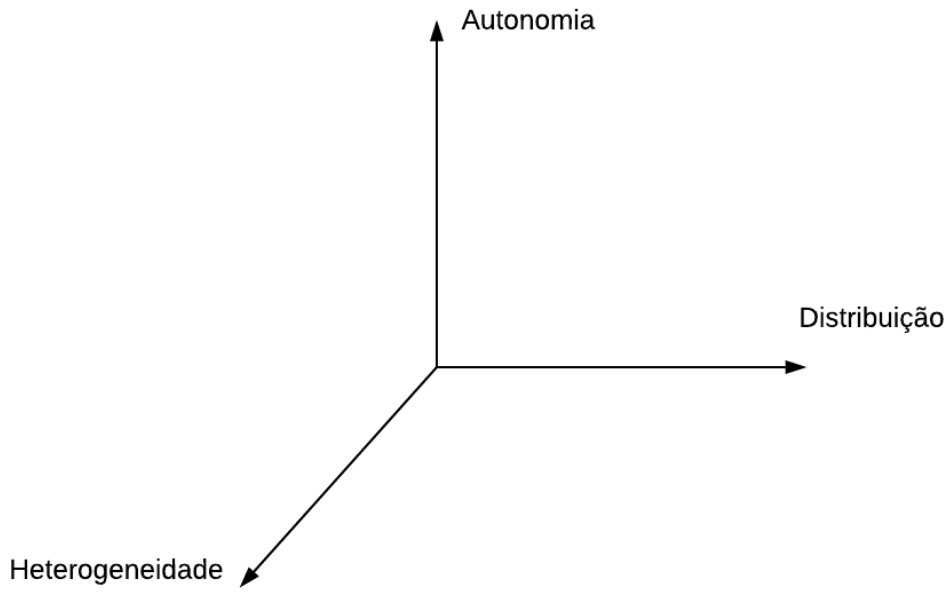

FIG. 2.6: Vetores de Interoperabilidade

Enquanto a distribuição está relacionada à localização dos dados nos mais diversos

ambientes integrados, a autonomia está relacionada à forma como as informações são gerenciadas pelos mais diversos entes. Para o autor, as entidades que gerenciam diferentes fontes de informação só estão dispostas a compartilhar seus dados se eles ainda permanecerem no controle de seu acesso. Logo, é importante entender os diversos aspectos da autonomia e como eles podem ser trabalhados quando um determinado sistema participa de uma federação ou compartilha seus dados com novos usuários ou aplicações.

Um componente participante de um sistema pode exibir vários tipos de autonomia: de projeto, de comunicação, associação e execução. A autonomia de projeto refere-se à habilidade do componente de escolher seu próprio projeto com respeito a qualquer coisa, como o dado ou informação sendo gerenciada (universo do discurso, domínio), a representação (modelo de dados, linguagem de consulta), o nome dos elementos contendo dados (ontologia usada), a conceitualização ou interpretação semântica dos dados, etc. Já a autonomia de comunicação refere-se à habilidade dos componentes de decidir se e quando se comunicam com outros componentes. Um componente com autonomia de comunicação é capaz de decidir quando e como responder a um pedido de outro componente. Finalmente, a autonomia de execução refere-se à habilidade de um componente executar operações locais sem a interferência de um ente externo e de decidir a ordem na qual as operações externas são executadas. Assim, um sistema externo não pode forçar uma ordem de execução de comandos e suas operações locais são logicamente isoladas pela participação em uma federação. Além disso, o componente não precisa informar um sistema externo ou federado quais operações externas são executadas e a ordem das operações externas em relação às operações locais. Ainda segundo Sheth, o componente exercita sua autonomia de execução tratando as operações externas da mesma maneira que as operações internas.

Em uma primeira análise, as heterogeneidades podem ser divididas em duas partes: a sistêmica e a informacional. A heterogeneidade sistêmica em sentido lato está relacionada às diferenças no *hardware*, no *software* e nos enlaces de comunicação dos sistemas (heterogeneidade de plataformas ou sistêmica em sentido estrito) e às diferenças relacionadas ao sistemas gerenciadores de banco de dados e suas capacidades (heterogeneidade dos sistemas de informação). Já a heterogeneidade informacional está ligada ao conteúdo que repousa nos mais diversos sistemas. Ao focar nesta dimensão crucial da heterogeneidade e em suas correspondentes soluções, a discussão apresenta diferentes níveis de interoperabilidade: a sintática, a estrutural e a semântica.

Na heterogeneidade sintática, o escopo é a diferença nos aspectos de capacidade de leitura dos dados por máquinas, conhecido também como problemas de formatação. Já na heterogeneidade estrutural, a preocupação repousa nos diferentes construtos de mo-

delagem para a representação dos dados. A heterogeneidade esquemática, que aparece particularmente em bases de dados estruturados, é também um aspecto da heterogeneidade estrutural. Finalmente, a heterogeneidade semântica está ligada à diferença de significado dos dados e dos metadados quando um sistema informacional é comparado com outro. A Figura 2.7 representa a classificação das heterogeneidades proposta em (SHETH, 1999).

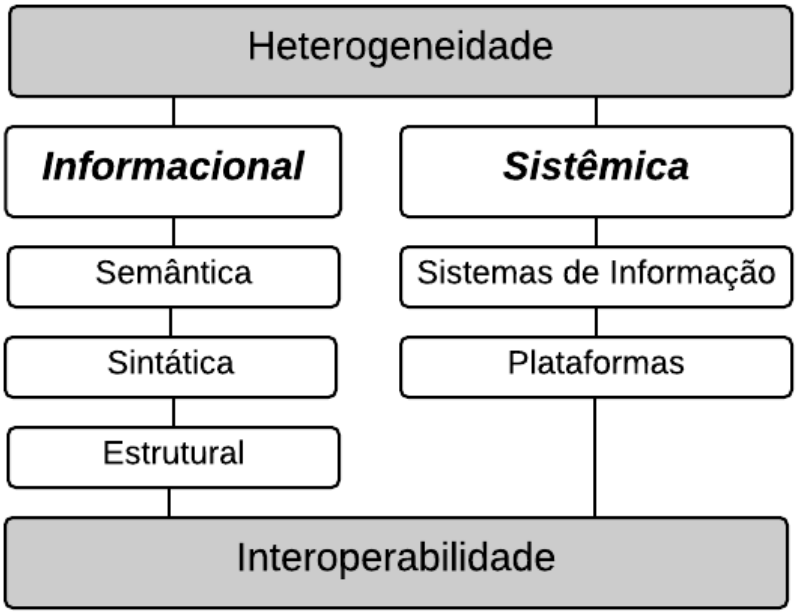

FIG. 2.7: Heterogeneidades e Interoperabilidades

2.2.2 ESTILOS DE INTEGRAÇÃO

De acordo com Hohpe e Woolf (2003), todas as soluções de integração precisam lidar com alguns desafios fundamentais: a confiabilidade e a latência das redes de comunicação, a diversidade de aplicações e a inevitabilidade das mudanças. Em relação a confiabilidade das redes de comunicação, já que as soluções de integração precisam transportar dados de um computador para outro, nota-se que em comparação a processos executados em um único aparato, aqueles configurados de forma distribuída precisam lidar com um conjunto muito maior de problemas. Invariavelmente, sistemas que precisam ser integrados estão separados por continentes e os dados necessitam viajar através de linhas telefônicas, redes locais, roteadores, *switches*, enlaces via satélite, entre outros dispositivos de comunicação, podendo provocar atrasos e interrupções em cada um desses passos. Já a latência da rede

de comunicação pode provocar a mudança no projeto da solução distribuída. Segundo os autores, enviar dados por meio da rede é várias ordens de grandeza mais demorado do que a chamada a um processo local. Sendo assim, projetar uma solução distribuída da mesma forma que uma aplicação local poderia ter implicações desastrosas relacionadas ao desempenho.

A diversidade das aplicações implica que a solução de integração seja capaz de realizar a interface entre diversos sistemas com naturezas tecnológicas diferentes. As soluções de integração precisam ser capazes de gerenciar a transferência de informações entre sistemas que possuem diferentes linguagens de programação, com diferentes plataformas operacionais e diferentes formatos de dados. Além disso, a mudança é uma regra quando se fala em aplicações, elas mudam ao longo do tempo. Uma solução de integração precisa manter o ritmo a despeito das mudanças que ocorrem nas aplicações que conecta, podendo ser facilmente pega em um efeito avalanche de modificações. Se um sistema muda, todos os outros podem ser afetados. A solução precisa então minimizar as dependências de um sistema com o outro, utilizando assim o conceito de baixo acoplamento.

Ao longo do tempo, os desenvolvedores superaram este desafios utilizando as seguintes abordagens:

Transferência de Arquivos : Uma aplicação escreve um arquivo para que outra possa ler. Estas aplicações precisam concordar não só com o nome do arquivo e sua localização, mas também com o seu formato, com o intervalo entre a sua escrita e leitura e sobre a responsabilidade pelo seu expurgo;

Banco de Dados Compartilhado : Múltiplas aplicações compartilham o mesmo esquema em um banco de dados, localizado fisicamente em um único repositório, não havendo assim duplicação. Dessa forma, nenhum dado precisa ser transferido de uma aplicação para outra;

Invocação Remota de Procedimentos : Uma aplicação expõe algumas de suas funcionalidades de tal forma que pode ser acessada remotamente por outras aplicações, como se fosse um procedimento local. A comunicação acontece de forma síncrona e em tempo real;

Mensageria : Uma aplicação publica uma mensagem em um canal comum destinado para este fim, enquanto outras a leem em qualquer instante, resultando em uma comunicação assíncrona. Para utilizar esta abordagem, as aplicações precisam concordar sobre o canal e sobre o formato da mensagem.

Embora as quatro abordagens solucionem o mesmo problema, cada estilo tem suas próprias vantagens e desvantagens. De fato, as aplicações podem ser integradas utilizando múltiplos estilos de tal forma a aproveitar o ponto forte de cada um.

2.3 INTEGRAÇÃO DE DADOS BASEADA NA CARACTERIZAÇÃO DAS FONTES DE DADOS

A utilização das características das fontes de dados para resolver os problemas da integração de dados tem sido abordada ao longo do tempo por outros trabalhos acadêmicos.

Em Cao et al. (2007), os autores propõem uma estrutura de integração de dados que possa dinamicamente ajustar seu esquema de integração toda vez que uma mudança é percebida no esquema de alguma das fonte de dados participantes do ambiente, utilizando para isso ontologias, similaridade semântica, *web services* e a conversão do modelo lógico do conteúdo para um modelo hierárquico compatível com o formato XML. Embora a conversão das fontes de dados seja uma solução possível para normalizar o modelo lógico dos conteúdos das fontes de dados participantes do ambiente de integração, essa transformação adicional pode não ser necessária para integrá-las. Além disso, esta operação aumenta a latência total de integração, o que pode não ser aceitável caso haja alguma limitação imposta pelos sistemas consumidores.

Já Khazankin e Dustdar (KHAZANKIN; DUSTDAR, 2010) argumentam que as aplicações utilizam cada vez mais dados de fontes *web* e que, apesar da escolha da técnica de integração seja tradicionalmente feita pelo analista ou pelo desenvolvedor, o aumento das fontes inviabiliza a administração do sistema pelo integrador, tendo que lidar com todas as variações de comportamento das fontes. Eles propõem a automatização das escolhas das técnicas de integração a partir de características da fonte de dados. Contudo, a proposta foca apenas em duas destas características: a latência e a largura de banda da rede. Michael Stonebraker, junto com outros autores (STONEBRAKER et al., 2013), propõem a utilização de um sistema de curadoria baseado na semântica dos dados. Para cada fonte nova, algoritmos de aprendizado de máquina analisam a fonte, identificam os atributos, agrupam-nos em tabelas e os transformam segundo critérios definidos. A intervenção humana só é necessária quando o sistema necessita de orientação para passar para os próximos estágios de integração. Apesar de ser um trabalho complexo e completo em relação à utilização de semântica para melhorar a integração de dados, outros aspectos, como as medições de qualidade da rede de comunicação, não são levados em conta.

Em Meisen et al. (2013), os autores propõem uma estrutura de integração adaptativa

TAB. 2.1: Tabela Comparativa dos Trabalhos Relacionados

		Cao et al. (2007)	Khazankin e Dustdar (2010)	Stonebraker et al. (2013)	Meisen et al. (2013)	**Proposta de Trabalho**
Heterogeneidade	Sintática	•				•
	Semântica	•		•	•	
	Estrutural				•	•
	Sistêmica		•			•
Abordagem de Integração	Materialização	•		•	•	
	Virtualização					
	Flexível		•			•

que seja capaz de lidar com as heterogeneidades estruturais e semânticas das fontes de dados em um ambiente de simulação na área de engenharia mecânica. Este trabalho procura lidar apenas com dois tipos de heterogeneidade: a estrutural e a semântica. As heterogeneidades sistêmica e sintática não são estudadas ou trabalhadas para resolver problemas relacionados à integração de dados.

Como se pode verificar, nenhum dos trabalhos relacionados utiliza integralmente as características das fontes de dados como descrito na classificação de Amit Sheth (SHETH, 1999). Da mesma forma, esta peça propõe-se a investigar e utilizar um subconjunto que pode contribuir para a seleção de abordagens de integração, almejando a minimização do tráfego de dados nas redes de comunicação e redução da intervenção humana no processo de integração o tanto quanto possível. A tabela 2.1 mostra o posicionamento do trabalho em relação aos anteriormente apresentados.

3 SELEÇÃO DE ABORDAGENS DE INTEGRAÇÃO POR MEIO DA CARACTERIZAÇÃO DAS FONTES DE DADOS

A pesquisa realizada sobre a caracterização de fontes de dados no contexto da integração de dados mostra que poucas publicações se dedicam a efetivamente relacioná-las, como em (KHAZANKIN; DUSTDAR, 2010). Geralmente, os trabalhos baseiam-se em ideias genéricas sobre a utilização de tais características ou apenas trabalham um enfoque, como a semântica dos atributos do conteúdo. Tornou-se então necessária uma revisão das características das fontes de dados, utilizando, principalmente, os trabalhos de classificação de Sheth (SHETH, 1999) e Hophe *et al* (HOHPE; WOOLF, 2003) como base importante na construção de uma sistematização. O objetivo deste capítulo é mostrar o levantamento realizado para identificar as características das fontes de dados que compõem um ambiente de integração, discutir a relevância de cada uma delas para a seleção de abordagens de integração e, por fim, desenvolver uma sistematização que possa auxiliar a escolha de tais abordagens.

3.1 LEVANTAMENTO DAS CARACTERÍSTICAS DE UM AMBIENTE DE INTEGRAÇÃO DE DADOS

Poucos trabalhos discutem detalhadamente sobre o uso das características das fontes de dados e de outros elementos do ambiente de integração para apoiar a seleção da abordagem de integração mais adequada. Cada um deles foca em um aspecto, seja da fonte de dados propriamente dita, seja do ambiente de integração para montar o arcabouço da solução que resolve o problema identificado. Como pode ser visto no levantamento bibliográfico realizado por de Faria Cordeiro (2015), vários autores propõem diversas classificações em relação às fontes de dados, muitas vezes utilizando nomes diferentes para explicar o mesmo conceito. Porém, nota-se que todos eles propõem classificações baseadas no artigo sobre interoperabilidade de sistemas de Sheth (1999). É importante também apontar que os trabalhos mais atuais no contexto da heterogeneidade de sistemas lidam mormente com o seu aspecto semântico. Amit Sheth já apontava este caminho em 1996, uma vez que, naquele momento, já se considerava que os problemas gerados pelas heterogeneidades sintática, estrutural e sistêmica já haviam sido resolvidos.

Para superar esta lacuna, foi realizado um levantamento amplo para determinar quais seriam as características e como elas poderiam interferir em qualquer aspecto do processo de integração. Para tanto, utilizou-se um ambiente genérico de integração de dados como ponto de partida para extração de características, sendo este representado na Figura 3.1.

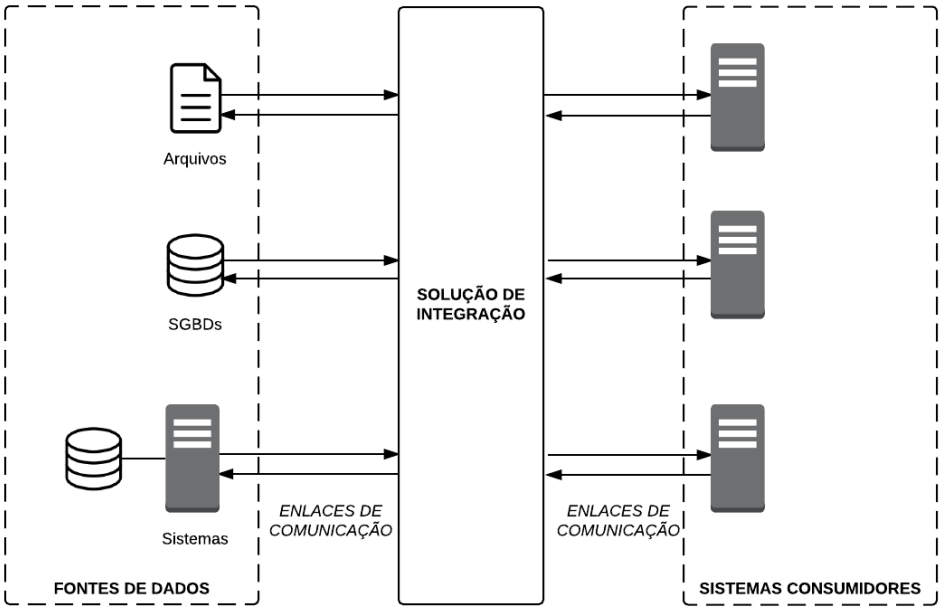

FIG. 3.1: Ambiente Genérico de Integração

Este ambiente de integração é inspirado do cenário descrito na Seção **??**. As fontes de dados são os diversos sistemas utilizados para gerenciar um sistema de telefonia celular. Nesse ecossistema de gerenciamento de redes, há uma grande diversidade de tecnologias sendo utilizadas (2G,3G,4G), formatos, tipos de repositórios e níveis de acesso. Cada sistema trata individualmente de suas tarefas, sendo que muitos deles, principalmente os designados para gerenciar tecnologias mais antigas como o 2G, não foram projetados para a troca de informações com outros sistemas. Os sistemas consumidores são *datawarehouses*, *datamarts* e CRMs que são responsáveis pela análise da qualidade da operação do sistema de telefonia celular, assim como pela bilhetagem e cobrança pelo uso do sistema por parte de sues clientes.

A solução de integração soluciona o *descasamento de impedância* encontrado entre as os sistemas de gerenciamento dos elementos de rede e os sistemas de análise e cobrança das operadoras de telefonia celular. Representa também um elemento de desacoplamento que aumenta a segurança no trato das informações e que evita que outros sistemas deteriorem seu desempenho ao periodicamente realizar consultas diretas aos seus dados, prejudicando

sua principal função que é prover acesso aos usuários da operadora e o controle dos elementos da rede de telefonia celular.

Ressalta-se ainda que a solução de integração apresentada possui um arquitetura híbrida ou de materialização parcial. Sendo assim, a solução de integração possui tanto tradutores (*wrappers*) quanto extratores (*extractors*), que podem ser construídos por tecnologias diferentes e utilizados indistintamente para aplicar as abordagens de virtualização e materialização. Nota-se também que este levantamento não só focou nas caraterísticas próprias das fontes de dados, mas também naquelas relacionadas ao ambiente em que estão inseridas.

3.1.1 CLASSIFICAÇÃO SEGUNDO A ABORDAGEM DE AMIT SHETH

Como se pode depreender do ambiente de integração apresentado, é possível separar suas características em quatro partes: as das *fontes de dados*, as da *solução de integração*, as dos *sistemas consumidores* e as dos *enlaces de comunicação* que os interconectam. A classificação de heterogeneidades de Sheth foi utilizada como ponto de partida do levantamento em cada uma dessas partes. Analisando a Figura 3.1, nota-se que as questões de heterogeneidade sintática, estrutural e semântica não se aplicam aos *enlaces de comunicação*. A questão da heterogeneidade sistêmica permeia todas as partes, uma vez que a própria definição dada por Sheth é ampla e engloba questões sobre *hardware* e *software* dos elementos do ambiente de integração, assim como aquelas relacionadas à comunicação entre eles.

A questão da heterogeneidade sintática está relacionada ao formato no qual um determinado conteúdo é representado. Neste caso, o problema é saber se o conteúdo é passível de ser lido por uma linguagem de máquina. Para o *sistema consumidor*, este não é um ponto de atenção, pois presume-se que só é possível guardar um conteúdo caso ele possa ser entendido por um artefato computacional. Porém, a questão para as *fontes de dados* e para a *solução de integração* é relevante. A solução de integração só é capaz de processar o conteúdo de uma determinada fonte de dados caso sua leitura seja possível. Considerando a arquitetura híbrida e a predileção pela abordagem de virtualização, uma vez que só traz o conteúdo da fonte de dados quando necessário, a primeira verificação deve ser feita em relação ao tradutor. Caso ele seja capaz de reconhecer o formato do conteúdo, será possível virtualizá-lo. Senão, será necessário avaliar se o extrator é capaz de reconhecer a sintaxe do conteúdo. Se nos dois casos não for possível o reconhecimento, o conteúdo não poderá ser integrado.

Já a heterogeneidade estrutural possui três aspectos que precisam ser estudados: a existência de um modelo lógico ou uma representação conhecida para o conteúdo, a compatibilidade dos esquemas e dos tipos dos metadados. A existência de um modelo lógico ou de uma representação conhecida do conteúdo é condição necessária para que seja considerada sua integração. Da mesma forma que na heterogeneidade sintática, um conteúdo só poderá ser virtualizado caso o tradutor seja capaz de reconhecer a representação do conteúdo. Em caso negativo, o mesmo processo de análise deverá ser realizado pelo extrator. Se ambos não forem capazes de realizá-lo, o conteúdo não poderá ser integrado. Seguindo a mesma lógica, a incompatibilidade de esquemas e de tipos de metadados são questões específicas a serem tratadas pela solução de integração e estão relacionadas à capacidade e ao tempo utilizado para transformar o conteúdo proveniente das fontes de dados em algo reconhecível pelos sistemas consumidores.

Por fim, a heterogeneidade semântica está ligada à incompatibilidade de significado entre os metadados das fontes de dados e dos sistemas consumidores. Assim como na heterogeneidade estrutural, a questão está na capacidade e no tempo de processamento da solução de integração para adequar o conteúdo das fontes de dados. As questões de incompatibilidade semântica possuem um espectro amplo de soluções, desde meros mapeamentos à utilização de ontologias para resolução de significados.

3.1.2 CLASSIFICAÇÃO SEGUNDO A ABORDAGEM DE HOPHE E WOOLF

Como dito anteriormente, a descrição de Amit Sheth sobre heterogeneidade sistêmica é ampla, tratando em um mesmo aspecto questões diversas sobre *hardware*, *software* e de comunicação entre sistemas. Realmente, estas diferenças foram sanadas ao longo do tempo, principalmente com o desenvolvimento de arquiteturas de rede largamente utilizadas (por exemplo, a arquitetura TCP/IP), o que permitiu a comunicação entre diferentes sistemas de forma transparente. Por conseguinte, este aspecto da classificação não contribuiu para a extração de características relevantes para o estudo.

Porém, dois outros trabalhos selecionados na revisão bibliográfica puseram uma nova luz sobre a caracterização no ambiente de integração de dados. Em White (2006), ao diferenciar as técnicas de integração, o autor coloca que certas fontes de dados precisam ser consultadas periodicamente para verificar se um conteúdo foi produzido, enquanto que outras possuem mecanismos que entendem que o conteúdo da fonte de dados foi alterado e o envia para a solução de integração. Na mesma direção, Hophe e Woolf abordam os estilos de integração nos capítulos iniciais do livro *Enterprise Integration Patterns*

(HOHPE; WOOLF, 2003). Para os autores, há quatro estilos principais de integração: a transferência de arquivos, o compartilhamento em bancos de dados comuns, a invocação remota de procedimentos e a mensageria. Não obstante outros aspectos relacionados a cada um dos estilos, um se destaca no escopo da caracterização de *fontes de dados*: o comportamento da fonte quando o seu conteúdo é atualizado.

Nos três primeiros estilos destacados por estes autores, a *solução de integração* precisa indagar as fontes para perceber se há mudança no conteúdo. Já no último estilo, a fonte de dados percebe uma mudança e a envia para a solução de integração. Sendo assim, pode-se dizer que as fontes dos três primeiros estilos possuem um comportamento passivo em relação à alteração do conteúdo, enquanto o último possui um comportamento ativo. Devido à natureza independente e assíncrona do último estilo, faz-se necessário que o conteúdo seja materializado para posterior processamento.

Outro aspecto derivado dos estilos de integração é a capacidade da fonte de dados responder a uma consulta. Em uma primeira análise, classificou-se as fontes por tipo: arquivos e sistemas. Uma consulta só seria possível em um sistema, enquanto que o mesmo não seria verdade para um arquivo. Contudo, existem tecnologias que hoje suportam a consulta em arquivos, o que já deixaria essa classificação comprometida. Sendo assim, para superar esta limitação de nomenclatura, coloca-se que as fontes de dados podem ter ou não capacidade de resposta a uma solicitação por outro sistema. Caso não a tenha, não será possível sua virtualização. Porém, diferente do comportamento da fonte de dados, que é um aspecto próprio da mesma, a capacidade de resposta está relacionada ao tradutor ou extrator sendo utilizado. Ou seja, a capacidade de resposta da fonte de dados está ligada à capacidade da solução de integração de colocar uma consulta para uma fonte de dados e lidar com o resultado retornado.

3.1.3 ABSTRAÇÃO DO CONCEITO DE FONTE DE DADOS

Uma vez levantados os aspectos estáticos mais promissores das fontes de dados e do ambiente de integração para a seleção da abordagem de integração mais apropriada, o próximo passo foi analisar e testar cada característica trazida. Para tanto, foram utilizados os diagramas de classe e de atividade da UML para consolidar ou refutar a relevância de cada um. É necessário salientar mais uma vez que estes artefatos foram desenvolvidos considerando que a solução de integração pode ser construída de tal maneira que é possível selecionar entre tradutores e extratores indistintamente. As Figuras 3.2 e 3.3 mostram os diagramas confeccionados.

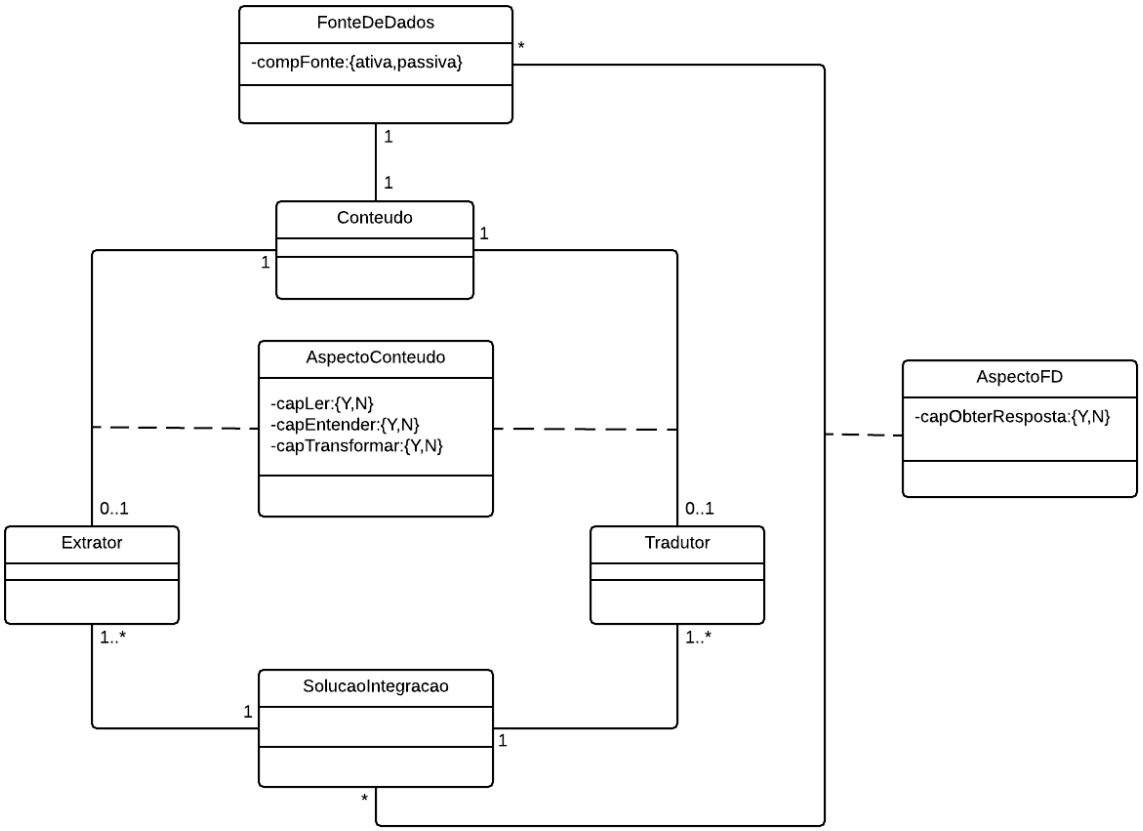

FIG. 3.2: Diagrama de Classe Conceitual - Análise de Aspectos Sintáticos, Estruturais e Semânticos

A partir do levantamento realizado nas Subseções 3.1.1 e 3.1.2 e nos diagramas apresentados (Figura 3.2 e 3.3), pode-se avaliar que:

– Ao analisar o comportamento e a capacidade de obter respostas das fonte de dados, nota-se que estes não são aspectos ligados ao seu conteúdo, mas sim à própria fonte de dados. A ideia por trás desta análise é que um conteúdo poderia ser transportado de uma *Fonte de Dados* para outra sem que suas características sintáticas, estruturais ou semânticas se perdessem. Assim, no diagrama de classe apresentado na Figura 3.2, a *Fonte de Dados* é representada distintamente de seu *Conteúdo* por meio do atributo *compFonte*, interligando-a a um único *Conteúdo*. Como colocado na Subseção 3.1.2, uma fonte de dados só pode ser virtualizada se seu comportamento for passivo. Caso contrário, a *Solução de Integração* precisa lidar com as atualizações dos conteúdos, sendo necessário a utilização de repositórios temporários para guardá-los para posterior processamento;

– Diferente do comportamento da fonte de dados (*FonteDeDados.compFonte*), que

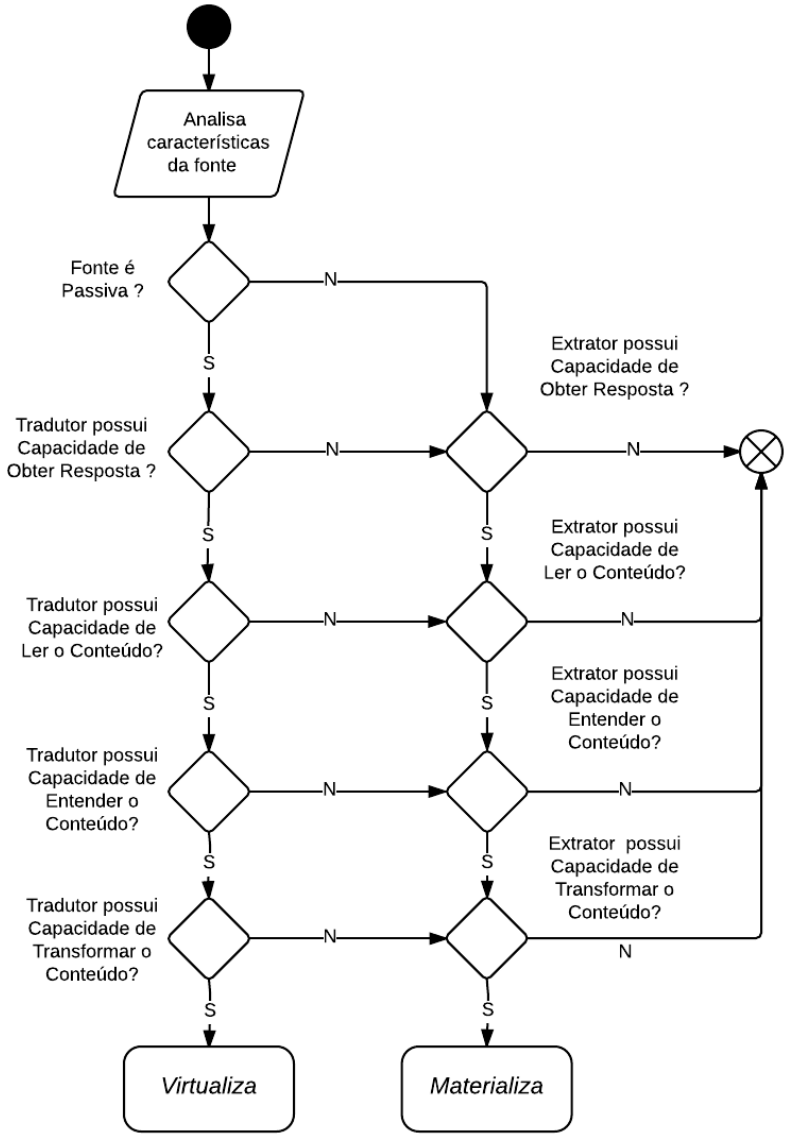

FIG. 3.3: Fluxo Decisório - Características Estáticas

não é influenciado pelas escolhas tecnológicas para a construção da solução de integração, a capacidade de obter respostas de uma determinada fonte de dados está diretamente ligada à capacidade dos tradutores e dos extratores de enviar consultas a uma fonte de dados e compreender o resultado recebido. Ou seja, a capacidade de obter respostas (*AspectoFD.capObterResposta*) é um atributo do relacionamento entre a *Fonte de Dados* e a *Solução de Integração*. Uma determinada *Fonte de Dados* só pode ser virtualizada caso a *Solução de Integração* possa enviar uma consulta em sua direção e ser capaz de processar o resultado de tal consulta;

– A existência de uma sintaxe e de uma representação do *Conteúdo* que possa ser

reconhecida por parte de um *Tradutor* é condição necessária para possibilitar a virtualização. Ou seja, o *Tradutor* precisa ser capaz de ler *(AspectoConteudo.capLer)* e entender *(AspectoConteudo.capEntender)* o conteúdo sendo integrado. Caso contrário, a mesma análise precisa ser realizada por um *Extrator* para que, pelo menos, o conteúdo seja integrado. Nota-se também que estes atributos são dos relacionamentos entre o *Conteúdo* e os *Tradutores* e *Extratores*;

– As incompatibilidades do modelo lógico, do domínio e da semântica dos metadados entre os esquemas das *Fontes de Dados* e os *Sistemas Consumidores* precisam ser resolvidas pela *Solução de Integração* por meio de manipulações do conteúdo (*AspectoConteudo.capTransformar*). Se o *Tradutor* não for capaz de resolvê-las, o conteúdo não poderá ser virtualizado. Se o mesmo ocorrer para um *Extrator*, o conteúdo não poderá ser integrado. Assim como os atributos *AspectoConteudo.capLer* e *AspectoConteudo.capEntender*, este atributo caracteriza também o relacionamento entre o *Conteúdo* e os *Tradutores* e *Extratores*.

Considerando a lista anterior, pode-se inferir que apenas o comportamento da fonte de dados é independente do relacionamento desta com a solução de integração. Enquanto isso, a capacidade de resposta da fonte de dados e a resolução das questões sintáticas, estruturais e semânticas de seu conteúdo são aspectos dos relacionamentos entre as fontes de dados e seus conteúdos em relação à solução de integração. Dessa forma, as escolhas tecnológicas para implementar os tradutores e extratores da solução de integração influenciarão na escolha da abordagem de integração mais apropriada para cada fonte de dados participante do ambiente.

Há de se ressaltar que o diagrama mostrado na Figura 3.2 exibe um conceito. Como colocado por (DOAN et al., 2012), a criação de tradutores pode ser realizada de forma manual, automática ou interativa, sendo que a primeira é a forma mais comum. Acontece de maneira análoga com os extratores, que possuem formas de construção menos padronizadas que os tradutores. Logo, a avaliação da capacidade do tradutor ou do extrator de lidar com um conteúdo de uma determinada fonte de dados, assim como do comportamento da mesma, é feita no momento da construção do próprio artefato computacional. Uma vez que o objetivo da solução de integração é trazer o conteúdo da fonte de dados somente quando necessário, é suficiente que, para os efeitos de **escolha dinâmica** das abordagens de integração, a mesma saiba que uma fonte de dados é passível de virtualização devido a forma como seus tradutores e extratores foram desenvolvidos. Logo, basta que esta condição seja representada como a ***capacidade de virtualização*** de uma determinada

fonte de dados, como mostra a Figura 3.4, sendo então um atributo do relacionamento entre a *Fonte de Dados* e a *Solução de Integração* (*AspectoEstatico.eVirtualizavel*).

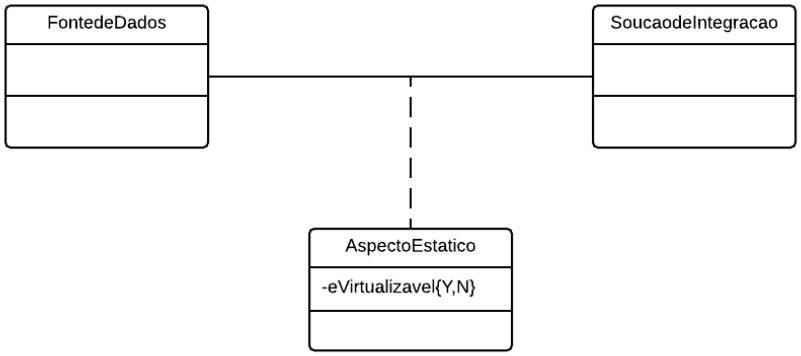

FIG. 3.4: Diagrama de Classe Conceitual - Síntese da Análise do Aspectos Estáticos

Porém, outros atributos podem frustrar uma abordagem de virtualização. Por exemplo, um tradutor pode ser capaz de ler, reconhecer e processar um conteúdo, porém o tempo para realizar tais ações pode frustrar a virtualização em virtude de requisitos não funcionais estabelecidos pelos sistemas consumidores. Neste sentido, outros aspectos de natureza dinâmica precisam ser avaliados. A próxima seção trata de tais características.

3.1.4 ANÁLISE DOS ASPECTOS DINÂMICOS

Apesar dos vários aspectos importantes trazidos e tratados por Sheth (1999) e Hohpe e Woolf (2003), nota-se que outros de natureza mais dinâmica não foram tratados, como o tempo de vida e o volume do conteúdo. Vale ressaltar que as alterações no modelo ou nos esquemas lógicos, que podem ocorrer ao longo do tempo de vida do ambiente de integração, não fazem parte desta análise devido a sua menor incidência no intervalo de tempo sendo analisado, que compreende apenas um ciclo completo de integração de uma determinada fonte de dados. Para superar essa lacuna, a lista abaixo representa uma primeira tentativa de relacionar as características dinâmicas nos ambientes de integração:

- **Tempo de Vida do Conteúdo**: é o tempo de permanência do conteúdo em um repositório;
- **Volume**: é o tamanho do conteúdo;
- **Frequência de Atualização do Conteúdo**: é o ritmo em que novos conteúdos são criados ou atualizados;

– **Frequência de Verificação dos Sistemas Consumidores**: É o ritmo no qual o sistema consumidor consulta a fonte de dados, procurando por novos dados a serem internalizados;

– **Tempo de Transporte do Conteúdo**: É o tempo necessário para transportar o conteúdo entre entes do ambiente de integração;

– **Tempo de Processamento**: É o tempo necessário para adequar o esquema da fonte de dados ao esquema do sistema consumidor (modelo lógico, domínio e significado dos atributos) e para aplicar as eventuais manipulações de seu conteúdo;

– **Banda da Rede de Comunicação**: É medida da capacidade do enlace que interliga os entes do ambiente de integração em bits por segundo;

– **Latência da Rede de Comunicação**: É o tempo levado entre o envio e o recebimento de um pacote na rede de comunicação. Dependente não só da banda, característica da rede de comunicação, mas também do número de elementos que estão na conexão dos entes do ambiente de integração como roteadores, *s*witches etc.

O volume e a frequência de atualização do conteúdo são características muito presentes e normalmente expostas quando a solução de integração está em um contexto *Big Data*. São, na verdade, dois dos aspectos mais aceitos para definir este tipo de ambiente. Em uma primeira análise, as duas características podem alterar o modo de lidar com o conteúdo. O volume pode alterar a forma de processamento (centralizado ou paralelo e distribuído) e a frequência de atualização pode alterar a forma de ingestão (em *batch* ou *stream*). Contudo, apenas a análise de suas definições mostrou-se insuficiente para determinar se a característica é relevante para apoiar a escolha de uma abordagem de integração. Sendo assim, alguns exercícios utilizando diagramas de sequência foram utilizados para melhorar a compreensão e extrair as características que definem um ambiente de integração de dados.

O diagrama de sequência mostrado na Figura 3.5 representa, de forma simplificada, a integração de um conteúdo de uma fonte de dados que foi materializada na solução de integração e, em seguida, internalizada pelo sistema consumidor. A solução de integração periodicamente verifica se um conteúdo está disponível. Quando ele o encontra, o extrai imediatamente. Em seguida, aplicam-se regras e transformações e o resultado é materializado em seu repositório. De maneira análoga, o sistema consumidor verifica

periodicamente se existe um conteúdo integrado a ser consumido no repositório da solução de integração e, quando o encontra, o internaliza em sua base de dados. É importante salientar que as duas operações de internalização do conteúdo são independentes neste contexto.

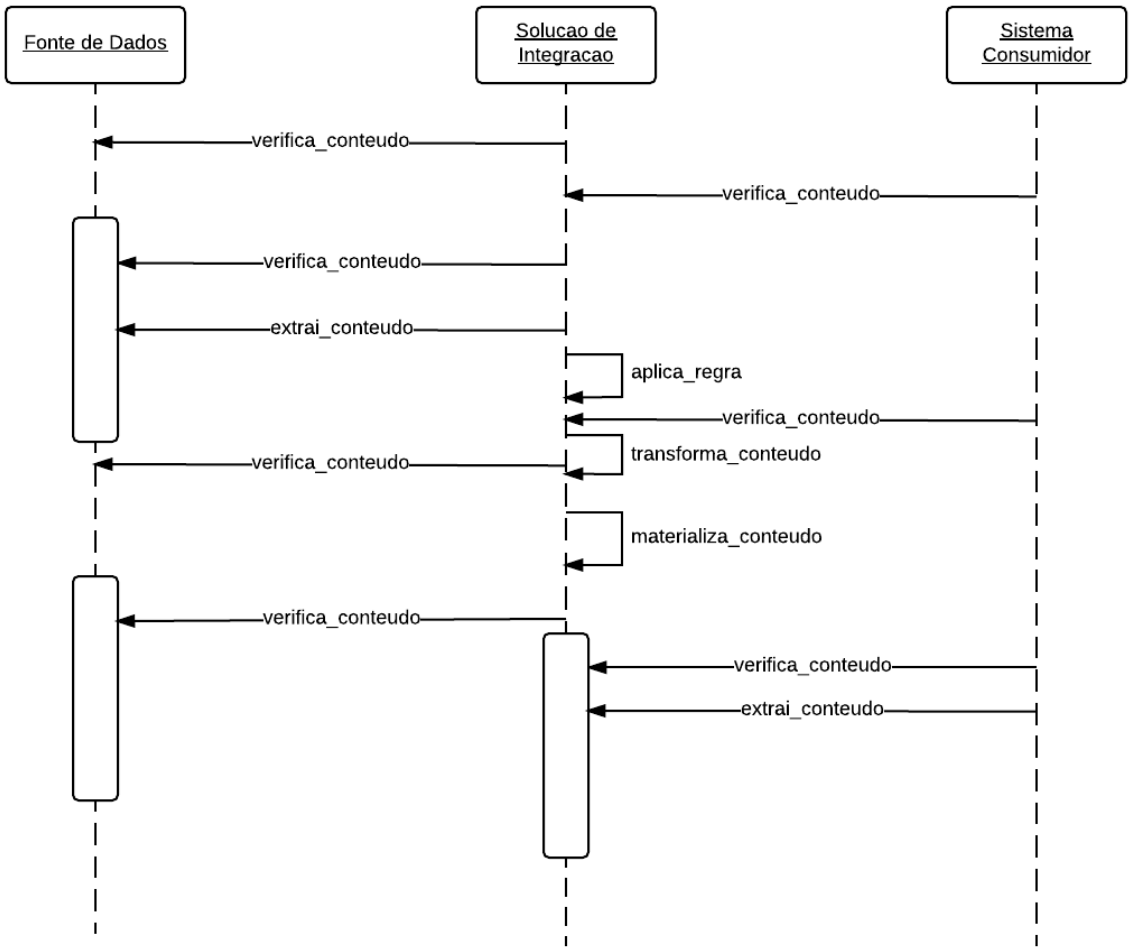

FIG. 3.5: Diagrama de Sequência - Internalização de um Conteúdo

Analisando o diagrama, o primeiro ponto a ser observado é a frequência ideal de verificação de existência de um conteúdo (f_V). Uma baixa frequência de verificação pode fazer com que um conteúdo seja perdido sem ter sido consumido. Por outro lado, uma alta frequência aumenta a necessidade de processamento por parte tanto das soluções de integração quanto dos sistemas consumidores, o que pode deteriorar suas operações. Em outras palavras, a frequência de verificação tem um limite superior (f_V^{max}) determinado pelo *hardware* e *software* destes mesmos sistemas e um limite inferior (f_V) tal que não permita a perda de um conteúdo a ser consumido.

Intuitivamente, percebe-se uma relação da frequência de verificação de conteúdo disponível com o tempo de vida do conteúdo (t_v) e sua frequência de atualização (f_a). Quanto menor o tempo de vida de um conteúdo, mais frequente deve ser a verificação de sua disponibilidade para integração. Caso contrário, é possível que o ente verificador perca a oportunidade de integrar um conteúdo de uma determinada fonte de dados. Por outro lado, quanto maior a frequência de atualização de um conteúdo, maior deve ser a frequência de verificação para evitar o acúmulo de conteúdos a serem integrados ao longo do tempo. O ambiente de integração deve estar preparado para lidar com este acúmulo caso não seja possível evitá-lo.

Embora a análise seja coerente, não é possível afirmar apenas com o discurso que estas relações podem afetar os aspectos do processo de integração e que, direta ou indiretamente, influenciem na escolha da abordagem de integração. Infelizmente, a revisão da literatura não mostrou qualquer investigação sobre estes relacionamentos e sua influência no processo de integração. Para superar este lapso, foram pesquisadas outras formas de obter e verificar a existência destes relacionamentos. Esta pesquisa deparou-se então com os experimentos clássicos na área de estatística e probabilidade, especialmente aqueles voltados a amostragens de sucesso ou falhas de um determinado experimento. A analogia percebida entre os experimentos clássicos e o problema de identificar a frequência ótima de verificação foi que os dois se baseiam em testes para determinar se um determinado resultado foi um sucesso ou uma falha. Colocando de outra forma, o experimento de classificar o resultado de um ensaio em sucesso ou falha é análogo à função de verificar se um conteúdo está ou não disponível para ser integrado ou consumido em um determinado intervalo de tempo.

Para facilitar o entendimento, pode-se utilizar o enunciado de um problema clássico de medição probabilística de sucesso como exemplo. Dada uma caixa com 10 bolas, sendo 6 pretas e 4 brancas, qual a probabilidade de se tirar um bola preta após três bolas quaisquer serem retiradas da caixa? A solução desta proposição é encontrada na distribuição binomial, que é largamente utilizada para caracterizar o resultado após a repetição de n ensaios independentes (ZENTGRAF, 2001). Esta distribuição determina a probabilidade de quantas vezes um determinado resultado acontece após a realização de n ensaios, sendo descrita pela fórmula abaixo :

$$p(x = k) = \binom{n}{k} p^k q^{n-k} \tag{3.1}$$

onde

n : tamanho da amostra,

p : probabilidade de sucesso de um ensaio,

q : probabilidade de falha de um ensaio,

k : quantidade de vezes que um resultado do ensaio (sucesso ou falha) é esperado,

$p(x = k)$: probabilidade de sucesso em k vezes em n ensaios

Utilizando esta formulação, chega-se à conclusão que a probabilidade de retirar uma bola preta após três bolas quaisquer serem retiradas é de aproximadamente 30% ($p(x = 1) = \binom{3}{1}(0.6)(0.4)^2 = 0.288$). A partir deste entendimento, a proposição clássica pode ser reescrita supondo que a necessidade agora é saber o número mínimo de ensaios a serem realizados, dada uma probabilidade de se tirar uma bola preta. É neste sentido que a analogia com este experimento clássico é usada para obter o relacionamento entre a frequência ideal de verificação (f_V), o tempo de vida do conteúdo (t_v) e a sua frequência de atualização (f_a). Para ajudar na sua compreensão, a Figura 3.6 representa o comportamento esperado do conteúdo de uma fonte de dados ao longo do tempo.

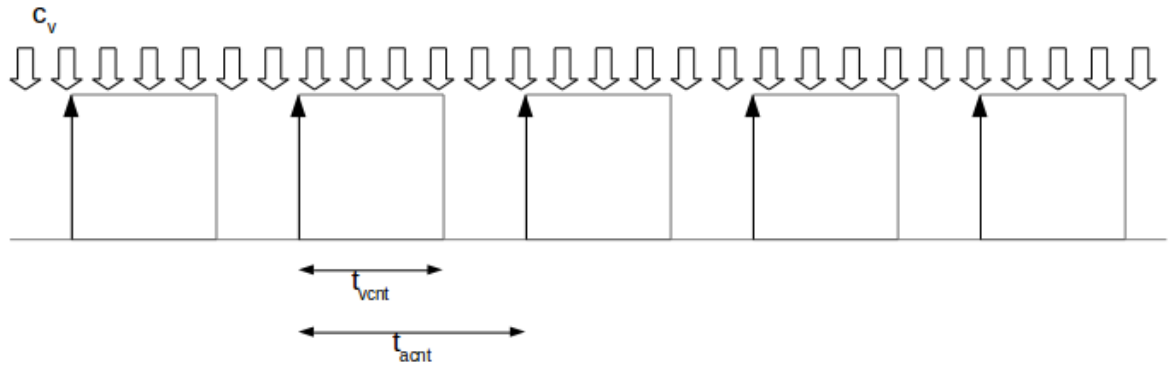

FIG. 3.6: Comportamento do Conteúdo ao Longo do Tempo

A frequência de verificação de conteúdo representa os ensaios a serem realizados para determinar a existência de conteúdo a ser processado (sucesso ou falha). A probabilidade de sucesso é a razão entre o tempo de vida do conteúdo e o tempo transcorrido entre suas atualizações ($t_a = 1/f_a$), enquanto a probabilidade de falha é o complemento da probabilidade de sucesso em relação a probabilidade total.

É importante salientar que, para este modelo, impõe-se que a verificação de um novo conteúdo ocorra dentro do intervalo de atualização da fonte de dados. Caso contrário, haverá um acúmulo de conteúdos para serem integrados ou consumidos, situação esta que deve ser administrada pelo ambiente de integração. Embora ela possa ser contornada

pelo aumento do repositório da solução de integração, pois está dentro do escopo de administração do ambiente, o mesmo pode não ser verdadeiro em relação aos repositórios das fontes de dados.

Utilizando a analogia com o experimento clássico e a Figura 3.6 como referência, observa-se que basta apenas um sucesso na amostragem para que o conteúdo possa ser extraído e processado. Ou seja, a frequência de verificação por um conteúdo possui uma frequência que permite sua identificação uma, duas ou até n vezes dentro do intervalo de atualização. O único resultado indesejável é a não identificação de um conteúdo a ser integrado ou consumido dentro do intervalo considerado. Pode-se então representar matematicamente a exclusão desta última hipótese pela probabilidade $p(x > 0)$. Porém, $p(x > 0)$ não é o mesmo que $p(x = 0)$, não sendo possível sua substituição direta na equação (3.1). No entanto, da teoria geral das probabilidades, é possível afirmar que $p(x > 0) + p(x \leqslant 0)$ representam o universo de todas as probabilidades, cuja soma é igual um. Esta identidade pode ser representada da seguinte forma:

$$p(x > 0) = 1 - p(x \leqslant 0) \tag{3.2}$$

Como os experimentos são discretos, é possível reescrever $p(x <= 0)$ por $p(x = 0)$, permitindo então sua substituição na equação (3.1). Depreende-se assim que quanto menor for $p(x = 0)$, maiores serão as chances de ocorrer pelos menos um sucesso na amostragem. Logo, substituindo k por zero na equação da distribuição binomial, tem-se que:

$$p(x = 0) = \binom{n}{0} p^0 q^{n-0} = q^n \tag{3.3}$$

Reescrevendo a equação anterior, utilizando as variáveis do problema mostrado na Figura 3.6:

$$p(x = 0) = q^n = (1 - t_v * f_a)^n \tag{3.4}$$

onde

n : quantidade de tentativas

t_v : tempo de vida do conteúdo,,

f_a : frequência de atualização do conteúdo,

$p(x = 0)$: probabilidade de não haver sucesso em n ensaios ou probabilidade de perda

Como o objetivo é encontrar o número mínimo de amostras necessárias (n), impondo a probabilidade de ocorrência de nenhum sucesso no experimento e calculando as probabilidades individuais de sucesso e falha, a equação (3.4) pode ser reescrita da seguinte forma:

$$n = \lceil \frac{\ln(p(x=0))}{\ln(1 - t_v * f_a)} \rceil^1 \tag{3.5}$$

Já que, por hipótese, os conteúdos são consumidos dentro do intervalo de atualização (t_a) e n representa a quantidade de vezes em que o conteúdo será investigado dentro deste intervalo, pode-se inferir então que a frequência de verificação é dada por:

$$f_V = n * f_a \tag{3.6}$$

Neste ponto, é necessário analisar uma inconsistência matemática introduzida pela utilização da função logarítmica para extrair o valor de n, que ocorre quando o tempo de vida do conteúdo (t_v) é igual ou superior ao intervalo de atualização do conteúdo (t_a). Esta condição cria logaritmos negativos ou nulo, o que é uma impossibilidade matemática. No entanto, ao analisar o tempo de vida do conteúdo em relação ao intervalo de sua atualização, nota-se que será sempre possível encontrá-lo disponível neste cenário. Colocando de outra forma, a probabilidade de encontrá-lo disponível dentro deste intervalo é de 100%, sendo então suficiente que a frequência de verificação (f_V) seja igual a pelo menos a frequência de atualização do conteúdo (f_a), ou seja, o mínimo de amostragens é dado por $n = 1$. Sendo assim, as equações (3.5) e (3.6) podem ser estendidas e melhoradas da seguinte forma:

$$n = \begin{cases} 1 & , \quad t_v \geqslant t_a \\ \lceil \frac{\ln(p(x=0))}{\ln(1 - t_v * f_a)} \rceil & , \quad t_v < t_a \end{cases} \tag{3.7}$$

$$f_V^{max} \geqslant f_V \geqslant n * f_a \tag{3.8}$$

O gráfico da Figura 3.7 abaixo mostra um exemplo de avaliação da frequência de verificação (f_V) a partir dos tempos de vida (t_v) e da frequência de atualização do conteúdo (f_a) e da probabilidade de perda do conteúdo para integração dentro do seu intervalo de atualização ($p(x = 0)$).

[1]Para manter a coerência, n é calculado como um arredondamento da equação (3.4), uma vez que representa uma quantidade de natureza discreta.

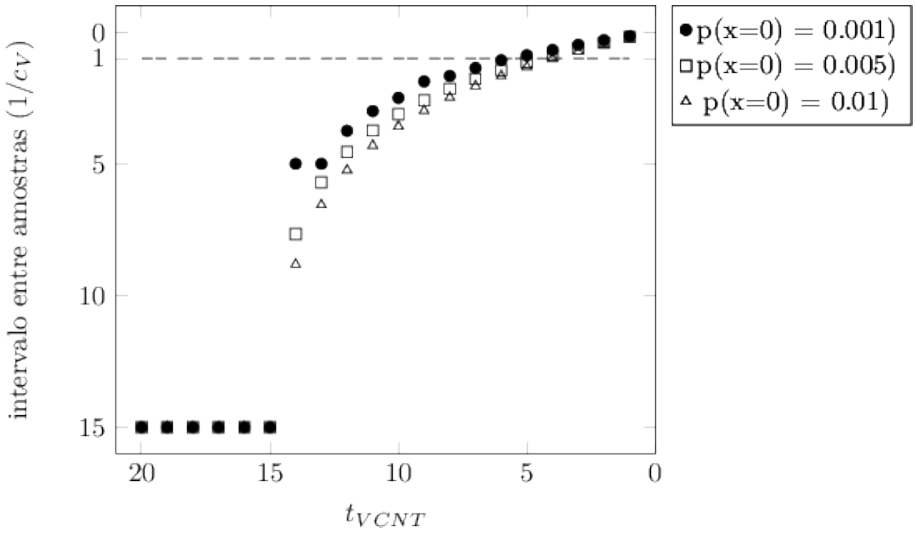

FIG. 3.7: Comportamento da Frequência de Verificação

O exemplo representa uma fonte de dados com uma frequência de atualização de quinze minutos e um tempo de vida de vinte minutos. A medida que o tempo de vida do conteúdo vai diminuindo de tal forma a ser inferior à frequência de atualização, a frequência de verificação vai aumentando até um máximo definido pelo *hardware* e *software* do ente verificador. O limite mostrado pela linha tracejada no gráfico é um exemplo do limite imposto por um sistema operacional, como o Windows Server 2012, onde a taxa mínima de execução do gerenciador de tarefas é de uma execução por minuto. Desta forma, haverá uma probabilidade de perda do conteúdo para integração dentro do seu intervalo de atualização superior à estipulada caso o tempo de vida do conteúdo da fonte de dados seja inferior a cinco minutos.

Mesmo resolvida a relação entre a frequência de atualização, a verificação e o tempo de vida de conteúdo, há uma outra situação limite que precisa ser explorada. Esta situação ocorre quando o ensaio encontra o conteúdo no limiar do seu tempo de vida. Uma vez que o transporte de um conteúdo entre entes do ambiente de integração não é instantâneo, é possível que o mesmo esteja sendo removido do seu repositório enquanto está sendo transportado. Para evitar esta ocorrência, retira-se este tempo necessário para transportar o conteúdo (t_T) do seu tempo de vida. Dessa forma, mesmo que o conteúdo seja percebido apenas no limiar de sua existência, será possível seu transporte sem se encontrar em um estado instável. Esta nova formulação é apresentada da seguinte maneira:

$$n = \begin{cases} 1 & , \quad (t_v - t_T) \geqslant t_a \\ \lceil \dfrac{\ln(p(x=0))}{\ln(1-(t_v - t_T) * f_a)} \rceil & , \quad (t_v - t_T) < f_a \end{cases} \tag{3.9}$$

Ainda analisando o tempo de transporte do conteúdo entre os entes do ambiente de integração, pode-se inferir que o mesmo é impactado tanto pelo volume quanto pela qualidade da rede de comunicação. Quanto maior o volume do conteúdo, maior o tempo necessário para transportá-lo. Em contra partida, quanto melhor a qualidade da rede de comunicação, menor o tempo necessário para realizar o transporte.

Apesar de medir o tempo de transporte do conteúdo entre os entes do ambiente de integração seja tecnicamente exequível, nem sempre será possível realizá-lo. Enquanto a medição do tempo de transporte entre as fontes de dados e a solução de integração está dentro do escopo de administração do ambiente, o mesmo não pode ser dito dos sistemas consumidores. Neste último caso, a medição deve ser feita nos sistemas consumidores, o que pode não ser viável ou permitido por seus administradores. Logo, para contornar eventuais restrições, o tempo de transporte pode ser estimado pela monitoração do volume de conteúdo transportado (v_c) e da latência de rede (t_L)(*round trip time*) entre a solução de integração e os demais entes do ambiente. Por conseguinte, o tempo de transporte do conteúdo pode então ser representado por $t_T = v_c * t_l$.

Outro ponto que merece atenção neste contexto é o tempo de processamento do conteúdo (t_p) para torná-lo compatível com os esquemas dos sistemas consumidores. Pode-se inferir que ele é dependente do volume do conteúdo (v_c) e da complexidade de sua transformação, que pode ser afetada por fatores relacionados às limitações de *hardware* e *software* onde reside a solução de integração. Diferente do volume, medir a complexidade de uma transformação de conteúdo parece uma tarefa mais árdua do que simplesmente medir o tempo consumido para realizá-la, que, no final, pode explicar todas estas considerações com apenas uma métrica.

Apesar da facilidade de monitoração do tempo de processamento de um determinado conteúdo, em comparação a características, é necessário investigar sua utilidade. Ao considerar a abordagem de materialização, o repositório de conteúdos integrados da solução de integração produz um isolamento entre as fontes de dados e os sistemas consumidores. Por conseguinte, para os sistemas consumidores, independe quanto tempo foi necessário para adequar o conteúdo. Este tempo está embutido no intervalo de atualização de conteúdo neste repositório (t_a^{SI}). Em contra partida, na abordagem de virtualização, como

não existe o repositório de conteúdos integrados, esta métrica é importante para avaliar se o tempo total entre a requisição às fonte de dados e sua efetiva disponibilização estão dentro dos requisitos não funcionais do ambiente.

Portanto, para o caso da materialização do conteúdo representado no diagrama de sequência do início desta discussão (Figura 3.5), os relacionamentos entre as características discutidas nesta seção podem ser descritos por meio das seguintes formulações:

$$n_{SI} = \begin{cases} 1 & , \quad (t_v^{FD} - t_T^{FDSI}) \geqslant t_a^{FD} \\ \lceil \dfrac{\ln(p(x=0)^{SI})}{\ln(1-(t_v^{FD} - t_T^{FDSI}) * f_a^{FD})} \rceil & , \quad (t_v^{FD} - t_T^{FDSI}) < t_a^{FD} \end{cases} \tag{3.10}$$

$$f_V^{SI} \geqslant n_{SI} * f_a^{FD} \tag{3.11}$$

$$n_{SC} = \begin{cases} 1 & , \quad (t_v^{SI} - t_T^{SISC}) \geqslant t_a^{SI} \\ \lceil \dfrac{\ln(p(x=0)^{SC})}{\ln(1-(t_v^{SI} - t_T^{SISC}) * f_a^{SI})} \rceil & , \quad (t_v^{SI} - t_T^{SISC}) < t_a^{SI} \end{cases} \tag{3.12}$$

$$f_V^{SC} \geqslant n_{SC} * f_a^{SI} \tag{3.13}$$

onde

t_v^X : tempo de vida do conteúdo em X, X $\in \{FD, SI\}$

f_a^X : frequência de atualização do conteúdo em X, X $\in \{FD, SI\}$

t_a^X : intervalo de atualização do conteúdo em X, X $\in \{FD, SI\}$

t_T^Z : tempo de transporte do conteúdo na interface Z, Y $\in \{FDSI, SISC\}$

f_V^X : frequência de verificação em X, X $\in \{FD, SI\}$

n_Y : quantidade de tentativas em Y, Y $\in \{SI, SC\}$

$p(x=0)^Y$: probabilidade de perda em Y, Y $\in \{SI, SC\}$

Ao analisar a interação entre a fonte de dados e a solução de integração (equações (3.10) e (3.11)), nota-se que a frequência de verificação da solução de integração (f_V^{SI}) só pode ser ajustada pelo tamanho da amostra (n_{SI}), uma vez que a frequência de atualização da fonte de dados (f_a^{FD}) não está, em tese, dentro do escopo de alteração a ser realizado pelo administrador do sistema de integração. Nos dois casos apresentados na equação

(3.10), apenas a probabilidade de perda ($p(x = 0)^{SI}$) é susceptível de ser alterada, uma vez que o tempo de vida do conteúdo, sua frequência de atualização e o tempo de transporte do conteúdo entre a fonte de dados e a solução de integração (t_T^{FDSI}) também estão além do escopo de administração do ambiente de integração.

Já no lado da interação entre a solução de integração e o sistema consumidor, apenas uma variável está indisponível para ajuste: o tempo de transporte do conteúdo integrado da solução de integração até o sistema consumidor (t_T^{SISC}). O tempo de vida do conteúdo integrado na solução de integração (t_v^{SI}) pode ser estendido com o aumento do repositório destinado para tal fim, enquanto que a frequência de atualização do conteúdo na solução de integração (f_a^{SI}) pode ser ajustado pela otimização do tempo de processamento do conteúdo na solução de integração (t_p), uma vez que $f_a^{SI} = 1/(t_T^{FDSI} + t_p)$. Já o ajuste da probabilidade de perda pode ser realizado da mesma forma vista na análise da interação anterior.

Uma vez terminada a análise para uma abordagem de materialização, faz-se necessário sua transposição para uma abordagem de virtualização para que seja possível a avaliação das características que podem interferir em sua escolha. No cenário de virtualização, a solução de integração é praticamente transparente para os sistemas consumidores, exercendo apenas a função de seletor entre as fontes de dados e os sistemas consumidores. O diagrama de sequência da Figura 3.8 representa este cenário.

Para a abordagem de virtualização, as equações (3.12) e (3.13) podem ser reescritas da seguinte maneira:

$$n_{SC} = \begin{cases} 1 & , \quad t' \geqslant t_a^{FD} \\ \lceil \dfrac{\ln(p(x=0)^{SC})}{\ln(1 - t' * f_a^{FD})} \rceil & , \quad t' < t_a^{FD} \end{cases} \tag{3.14}$$

$$t' = t_v^{FD} - t_T^{FDSI} - t_T^{SISC} - t_p$$

$$f_V^{SC} \geqslant n_{SC} * f_a^{FD} \tag{3.15}$$

onde

n_{SC} : quantidade de tentativas no sistema consumidor,

f_V^{SC} : frequência de verificação do sistema consumidor,

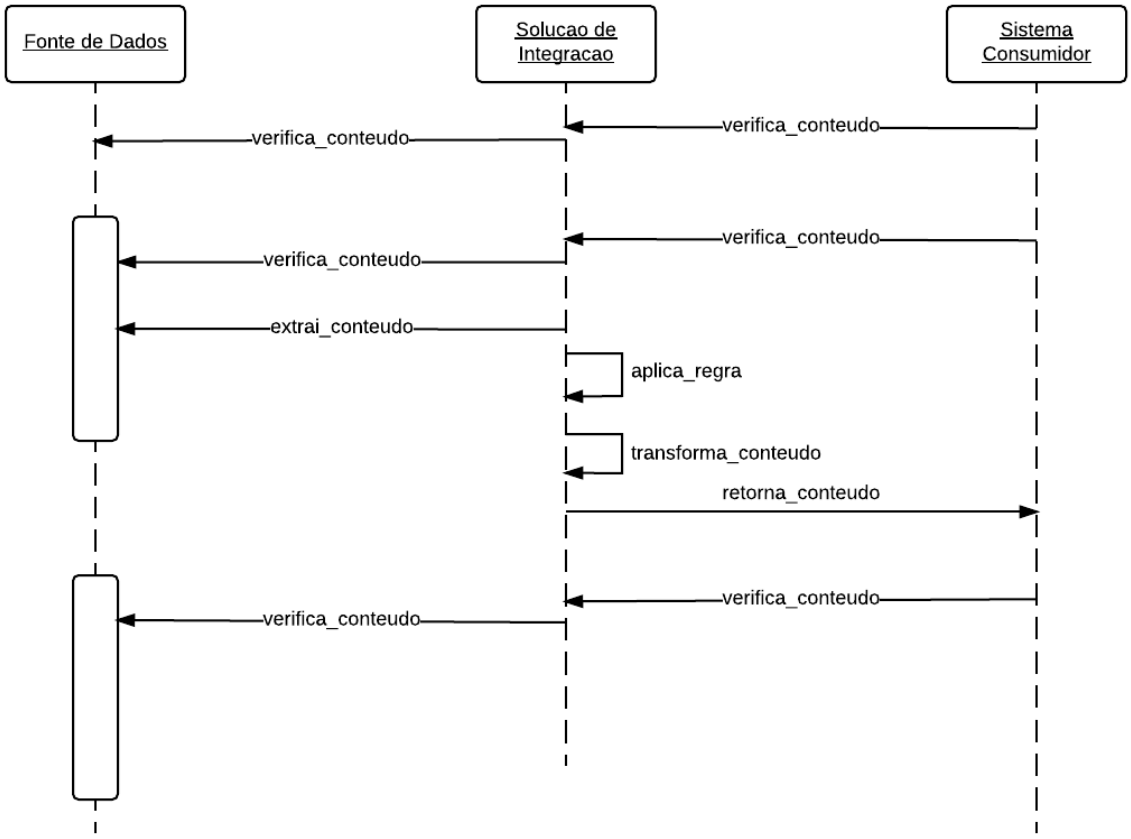

FIG. 3.8: Diagrama de Sequência - Virtualização de um Conteúdo

t_v^{FD} : tempo de vida do conteúdo na fonte de dados,

t_T^{FDSI}: tempo de transporte do conteúdo entre a fonte de dados e a solução de integração,

t_T^{SISC}: tempo de transporte do conteúdo entre a solução de integração e o sistema consumidor,

t_p^{SI} : tempo de processamento do conteúdo na solução de integração,

f_a^{FD} : frequência de atualização do conteúdo na fonte de dados,

$p(x = 0)^{SC}$: probabilidade de não haver sucesso em n ensaios ou probabilidade de perda

Neste ponto, faz-se necessária uma análise mais detalhada das opções mostradas na (3.14) para determinar as condições em que a virtualização pode ser utilizada como abordagem de integração. No caso onde $t^{'}$ é superior ou igual ao intervalo de atualização do conteúdo da fonte de dados (t_a^{FD}), o tamanho da amostra (n_{SC}) pode ser mínimo, ou seja,

a frequência de verificação do sistema consumidor deve ser no mínimo igual a frequência de atualização da fonte de dados. Neste caso, a fonte de dados pode ser virtualizada.

Caso contrário, a segunda opção precisa ser avaliada em conjunto com a equação (3.8). Esta combinação pode ser escrita da seguinte forma:

$$f_V^{SC,max} \geqslant f_V^{SC} \geqslant n_{SC} * f_a^{FD}$$

$$f_V^{SC,max} \geqslant f_V^{SC} \geqslant \lceil \frac{\ln(p(x=0)^{SC})}{\ln(1 - t' * f_a^{FD})} \rceil * f_a^{FD}$$

Observando a última expressão, caso ela seja cumprida, a fonte de dados em questão pode ser virtualizada. Porém, há poucos ajustes que podem ser feitos. Neste caso, somente a probabilidade de perda ($p(x = 0)$) pode ser ajustada e o tempo de processamento (t_p) pode ser otimizado, uma vez que as outras variáveis estão fora do escopo de administração do ambiente de integração. Em última análise, a frequência de verificação do sistema consumidor (f_V^{SC}) também pode ser ajustada até o seu limite ($f_V^{SC,max}$) para permitir a virtualização. Porém, esta última opção depende da interação entre os administradores da solução de integração e do sistema consumidor. Se nenhum desses ajustes cumprir com o proposto na equação (3.14), a fonte de dados precisará ser materializada. O fluxo decisório mostrado na Figura 3.9 exibe o resultado das últimas análises para determinar, a partir das características dinâmicas relacionadas, a abordagem de integração mais adequada.

3.2 SELEÇÃO DE ABORDAGENS DE INTEGRAÇÃO

O próximo passo realizado, após levantamento das características mais comuns encontradas nos ambientes de integração, foi a criação de uma matriz de interdependência para identificar se o comportamento de uma determinada característica poderia explicar o comportamento de outra. O objetivo desta matriz é prover uma revisão das análises anteriormente realizadas e extrair o conjunto mínimo de características que possam auxiliar a escolha da abordagem de integração mais apropriada para uma determinada fonte de dados. O resultado é mostrado na tabela 3.1:

De posse de todo o arcabouço gerado, as análises são sintetizadas da seguinte forma :

– Analisando os aspectos do ambiente de integração a partir das classificações de Sheth e Hophe, nota-se que, para a seleção dinâmica da abordagem de integração mais adequada para uma determinada fonte de dados, faz-se necessário apenas avaliar se

TAB. 3.1: Matriz de Interdependência

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Sintaxe															
2	Modelo Lógico															
3	Domínio dos Atributos															
4	Significado dos Atributos															
5	Complexidade de Transformação		•	•	•											
6	Comportamento da Fonte															
7	Capacidade de Resposta															
8	Ciclo de Vida do Conteúdo															
9	Volume															
10	frequência de Atualização do Conteúdo															
11	frequência de Verificação dos Sistemas Consumidores															
12	Tempo de Transporte do Conteúdo									•					•	•
13	Tempo de Processamento		•	•	•					•						
14	Banda da Rede de Comunicação															
15	Latência da Rede de Comunicação														•	

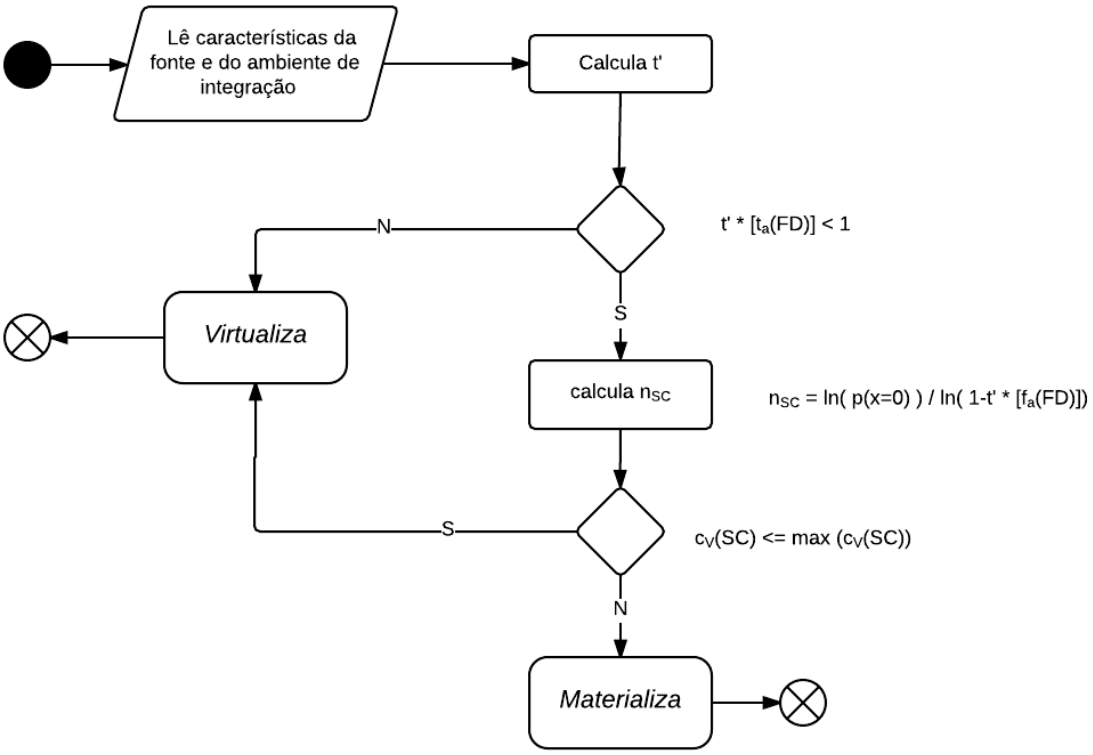

FIG. 3.9: Fluxo Decisório - Características Dinâmicas

a mesma é passível de virtualização a partir das escolhas de construção de tradutores e extratores no desenvolvimento da solução de integração;

– Medir a complexidade da transformação parece algo mais desafiador do que medir o tempo gasto para efetivamente executar a transformação. Ademais, o que pode ser complexo para uma solução de integração, pode não ser para outra. Como visto na matriz de interdependência (tabela 3.1), o tempo de processamento não só pode explicar a complexidade de transformação como também a variação do volume do conteúdo. Além disso, utilizar o tempo de processamento também explica as limitações de *hardware* e *software* da solução de integração ;

– Diferente do tempo de processamento gasto pela solução de integração para adequar o esquema das fonte de dados, medir o tempo de transporte do conteúdo pode não ser factível. Enquanto o tempo de transporte do conteúdo é passível de ser medido entre a fonte de dados e a solução de integração, o mesmo não pode ser dito em relação ao transporte entre a solução de integração e o sistema consumidor. Tudo está relacionado ao escopo de administração do ambiente de integração. De forma conservadora, presume-se que o escopo de administração se resuma à solução de

integração. Sendo assim, é importante monitorar o volume e a latência do enlace para se estabelecer uma estimativa do tempo de transporte. A medição destas duas características independe do escopo de administração do ambiente de integração;

– O tempo de vida do conteúdo (t_v^{FD}) assim como sua frequência de atualização (f_a^{FD}) são parâmetros necessários para determinar a escolha da abordagem de integração, assim como o tempo de seu transporte entre os entes do ambiente de integração (t_T^{FDSI} e t_T^{SISC}) , o tempo de processamento (t_p) para adequá-lo aos esquemas dos sistemas consumidores e a probabilidade de perda ($p(x = 0)^{SC}$) admitida pelo sistema consumidor.

Considerando a arquitetura idealizada na Seção 3.1 e a análise da relevância das características das fonte de dados, assim como das do ambiente de integração para a seleção da abordagem de integração mais apropriada para uma fonte de dados, percebe-se que as entidades participantes deste ambiente possuem dois papéis claros: a de *Produtores* e a de *Consumidores* de conteúdo. A Figura 3.10 mostra uma proposta de representação deste contexto utilizando um diagrama de classe conceitual.

Os papéis das *Fontes de Dados* e dos *Sistemas Consumidores* são claros nesta concepção: o primeiro é um *Produtor* de conteúdo enquanto o segundo é um *Consumidor* de conteúdo. Porém, o papel da *Solução de Integração* é menos evidente. Visto pela *Fonte de Dados*, a *Solução de Integração* é um *Consumidor* de conteúdo. Porém, quando visto pelos *Sistemas Consumidores*, a solução é um *Produtor* de conteúdo. Este duplo papel assumido pela *Solução de Integração* dificulta a representação no diagrama de classe, uma vez que a entidade não pode herdar as características de *Produtor* e *Consumidor* simultaneamente. Para contornar tal dilema, recorreu-se ao padrão de projeto apresentado por Fowler (1997), chamado de *Role Class Pattern*, que transforma o papel exercido na associação entre classes em uma classe específica. Tal solução é mostrada na Figura 3.11.

Uma vez estabelecido o duplo papel da *Solução de Integração*, faz-se necessário avaliar os possíveis estados da classe *Conteúdo*. Quando a classe *Produtor* é instanciada como uma *Fonte de Dados*, o estado de seu *Conteúdo* não é de interesse do ambiente de integração, uma vez que não interessa se o mesmo está materializado ou virtualizado. Interessa apenas se o mesmo está disponível para integração. Contudo, o mesmo não pode ser dito quando a classe *Produtor* é instanciada como uma *Solução de Integração*, pois isto determinará o momento em que o conteúdo da fonte de dados será transportado para o sistema consumidor.

Assim, a classe *Conteúdo* possui dois estados possíveis: *Materializado* e *Virtualizado*.

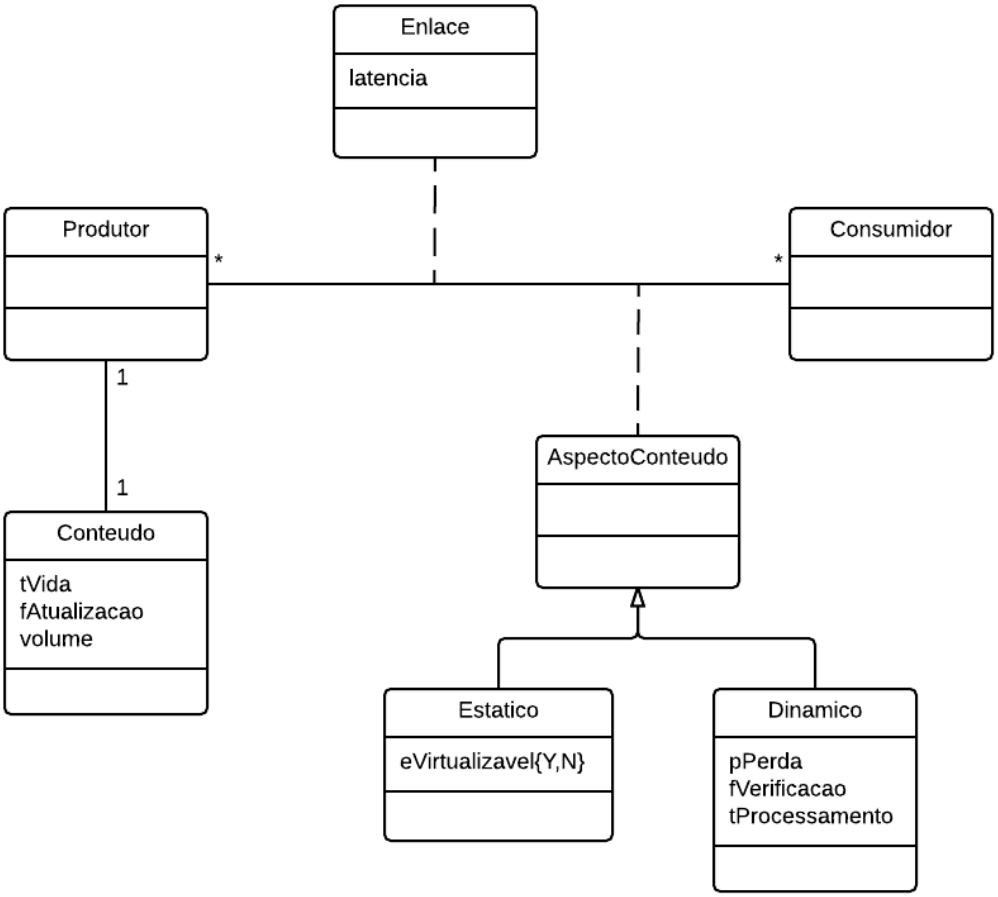

FIG. 3.10: Diagrama de Classe Conceitual - Entidades Produtoras e Consumidoras de Conteúdo

A passagem do estado inicial *Materializado* para *Virtualizado* é definida pelos aspectos estáticos e dinâmicos do relacionamento entre a *Solução de Integração* e a *Fonte de Dados* e é acionada pelo método *SolucaodeIntegracao.analisarAbordagem*, como mostrado na Figura 3.11. Do lado estático, é necessário que o atributo *AspectoEstatico.eVirtualizavel* seja verdadeiro, escolha esta realizada no momento da construção do processo de integração. Além disso, os aspectos dinâmicos devem permitir que a frequência de atualização da fonte de dados (f_a^{FD}) seja inferior que a frequência de verificação do sistema consumidor (f_v^{SC}), como mostrado na equação(3.1.4). Caso esta relação não permaneça, o *Conteudo* retornará ao estado *Materializado* após a execução do método *SolucaodeIntegracao.analisarAbordagem*. O diagrama de transição de estados resultante desta análise é mostrado na Figura 3.12.

Finalmente, o exemplo a seguir foi criado para ilustrar fluxo decisório apresentado na Figura 3.9 e o diagrama de eventos apresentado na Figura 3.9. Nele, fontes de dados

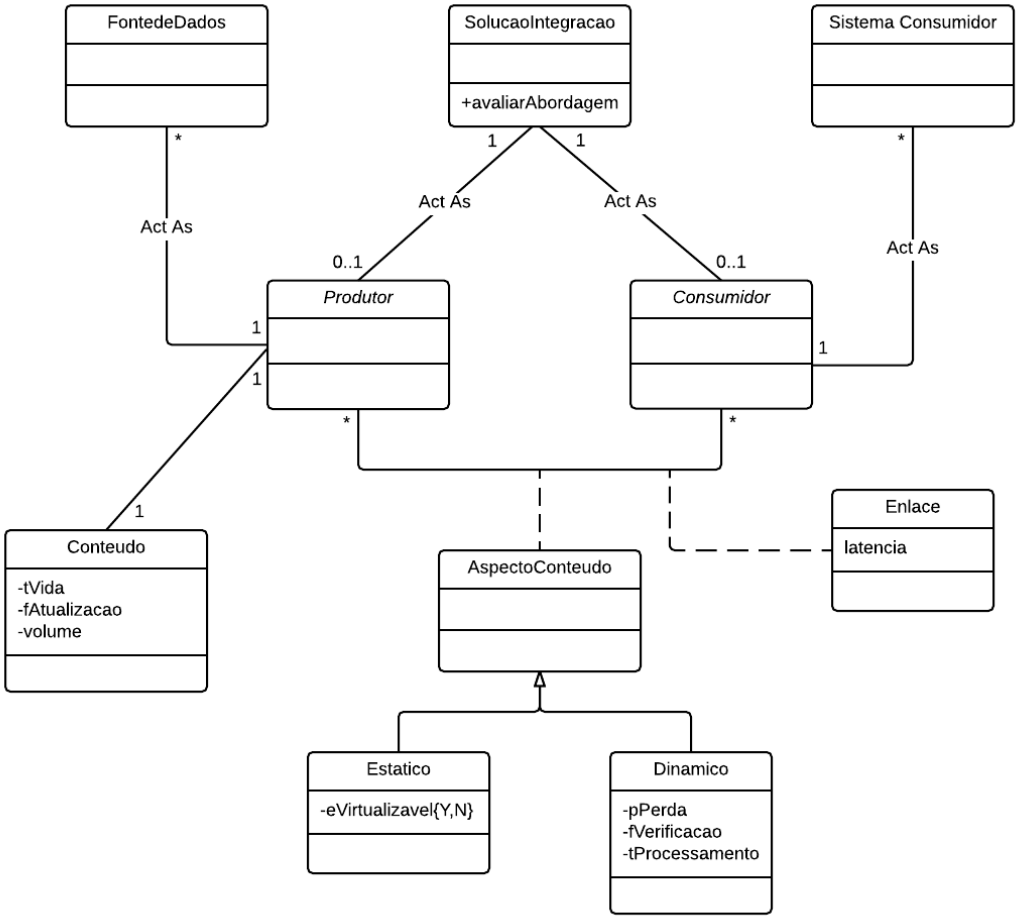

FIG. 3.11: Diagrama de Classe Conceitual - Representação do Ambiente de Integração Utilizando o Padrão de Projeto *Role Class*

fictícias foram submetidas ao fluxo de decisão utilizando valores típicos das características levantadas. Todas possuem uma linha de base que é mostrada na tabela 3.2. A fonte de dados A possui uma alteração em uma característica estática que impede sua virtualização, enquanto na fonte de dados B o tempo de vida do conteúdo é reduzido em um minuto a cada rodada de integração até o mínimo de quinze minutos e depois retorna ao seu valor original no fim da nonagésima quinta rodada. O mesmo é realizado com a fonte de dados C, porém a alteração é realizada na frequência de atualização do conteúdo da fonte de dados, chegando ao máximo de uma execução por minuto e depois retornando ao seu valor original ao fim da última rodada de integração. Finalmente, a última fonte de dados possui a alteração no tamanho do conteúdo na fonte de dados, iniciando com volume de 3 MB, sendo incrementado de 200 KB a cada rodada de integração até o final do processo. Neste caso, o tempo de processamento e o volume do conteúdo integrado também são alterados proporcionalmente ao aumento do volume do conteúdo da fonte de dados. As

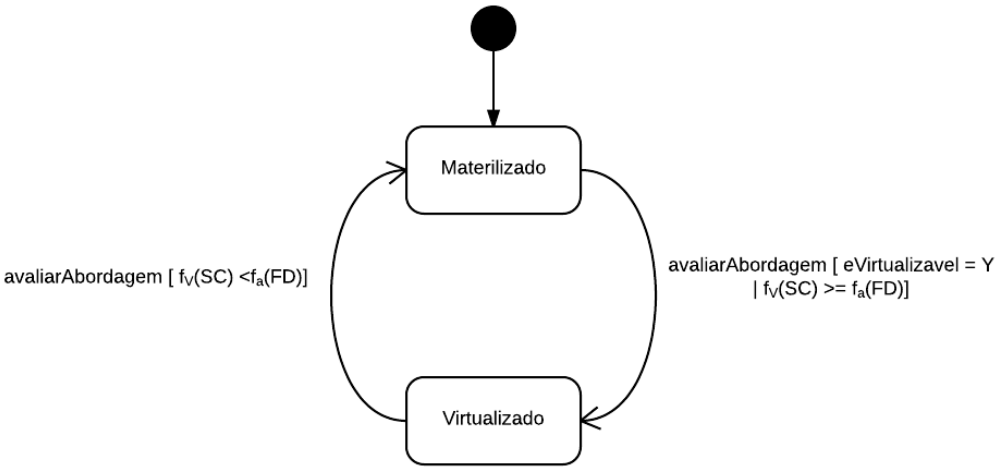

FIG. 3.12: Diagrama de Transição de Estados - Classe *Conteúdo*

TAB. 3.2: Linha de Base

Comportamento Ativo ?	Capacidade de Resposta ?	Existe Sintaxe ?	Existe Modelo Lógico ?	t_v^{FD}	f_a^{FD}	v_c^{FD}	t_l^{FDSI}	t_p	v_c^{SI}	t_l^{SISC}	p(x=0)	n_{SC}	f_V^{SC}
S	S	S	S	60	1/30	3000	10	60	2850	5	0,01	1	1/30

tabelas 3.3, 3.4 e 3.5 mostram extratos destas rodadas de integração e a Figura 3.13 mostra o comportamento de seleção da abordagem de integração. Para simplificação dos exemplos, foi colocada uma condição de troca de abordagem somente quando a indicação da possibilidade de uma nova abordagem é apresentada por pelo menos 3 vezes seguidas.

TAB. 3.3: Extrato Valores Típicos - Fonte de Dados B

Rodada	t_v^{FD}	f_a^{FD}	v_c^{FD}	t_l^{FDSI}	t_p	v_c^{SI}	t_l^{SISC}	p(x=0)	n_{SC}	f_V^{SC}	abordagem atual	abordagem possível	nova abordagem
1	60	1/30	3000	10	60	2850	5	0,01	1	1/30	M	V	M
2	59	1/30	3000	10	60	2850	5	0,01	1	1/30	M	V	M
3	58	1/30	3000	10	60	2850	5	0,01	1	1/30	M	V	M
4	57	1/30	3000	10	60	2850	5	0,01	1	1/30	M	V	V
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

TAB. 3.4: Extrato Valores Típicos - Fonte de Dados C

Rodada	t_v^{FD}	f_a^{FD}	v_c^{FD}	t_l^{FDSI}	t_p	v_c^{SI}	t_l^{SISC}	p(x=0)	n_{SC}	f_V^{SC}	abordagem atual	abordagem possível	nova abordagem
1	60	1/30	3000	10	60	2850	5	0,01	1	1/30	M	V	M
2	60	1/29	3000	10	60	2850	5	0,01	1	1/30	M	V	M
3	60	1/28	3000	10	60	2850	5	0,01	1	1/30	M	V	M
4	60	1/27	3000	10	60	2850	5	0,01	1	1/30	M	V	V
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

TAB. 3.5: Extrato Valores Típicos - Fonte de Dados D

Rodada	t_v^{FD}	f_a^{FD}	v_c^{FD}	t_l^{FDSI}	t_p	v_c^{SI}	t_l^{SISC}	p(x=0)	n_{SC}	f_V^{SC}	abordagem atual	abordagem possível	nova abordagem
1	60	1/30	3000	10	60	2850	5	0,01	1	1/30	M	V	M
2	60	1/30	3200	10	64	3040	5	0,01	1	1/30	M	V	M
3	60	1/30	3400	10	68	3230	5	0,01	1	1/30	M	V	M
4	60	1/30	3600	10	72	3420	5	0,01	1	1/30	M	V	V
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

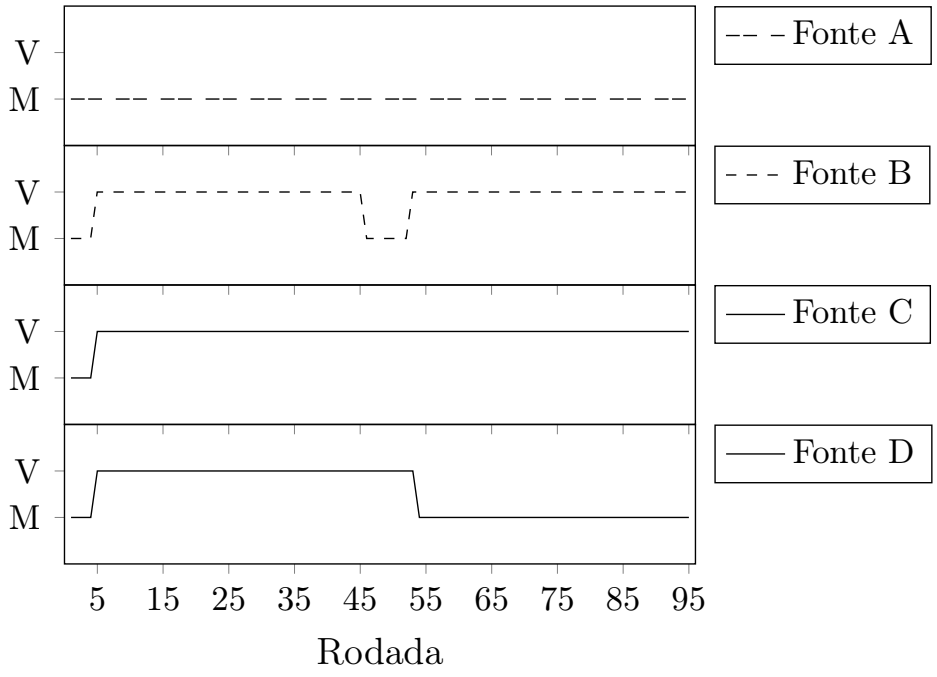

FIG. 3.13: Avaliação da Seleção de Abordagens de Integração

4 FLEXDI: UMA ARQUITETURA HÍBRIDA DE INTEGRAÇÃO DE DADOS

Uma vez estabelecido um método para a seleção de abordagens de integração por meio das características tanto das fontes de dados quanto do ambiente em que estão inseridas, faz-se necessário que a solução de integração seja capaz de lidar com tal possibilidade de variação ao longo do tempo, procurando assim minimizar a intervenção do administrador e o tráfego de dados nos enlaces de comunicação dos entes participantes. Para tanto, a solução de integração necessita:

a) monitorar as características das fontes de dados e do ambiente de integração para a seleção da abordagem de integração mais apropriada, conforme estabelecido na Seção 3.2;

b) selecionar, ao longo do tempo, a abordagem de integração mais apropriada para cada fonte de dados;

c) aplicar as manipulações de conteúdo (regras e transformações), independente da abordagem de integração sendo utilizada;

d) permitir o consumo dos conteúdos das fontes de dados pelos sistemas consumidores de forma transparente e única.

O objetivo deste capítulo é descrever a arquitetura da solução de integração, discutir as possíveis soluções para atender os requisitos anteriormente citados e as decisões tecnológicas tomadas para implementá-la.

4.1 REQUISITOS DA SOLUÇÃO DE INTEGRAÇÃO

A solução de integração é responsável pelo casamento de impedâncias entre as fontes de dados e os sistemas consumidores. Ou seja, ela é responsável pela adequação do conteúdo residente nas fontes de dados para que seja possível sua utilização pelos sistemas consumidores, sendo que este objetivo deve ser cumprido independente da abordagem de integração sendo utilizada. Logo, outros detalhes precisam ser observados na construção

da solução da integração além da simples implementação da seleção das abordagens de integração, conforme colocado no Capítulo 3.

O primeiro ponto a ser observado é a possibilidade que um determinado conteúdo tenha sido materializado em um período ao longo do tempo de vida do ambiente de integração, mas que, devido à mudança de suas características, a abordagem tenha mudado e passe a ser virtualizado. Considerando também que o mesmo pode ocorrer em sentido contrário, há a possibilidade que versões diferentes do conteúdo de uma determinada fonte de dados possam estar em repositórios diferentes em um instante do tempo de vida do ambiente de integração. Como um dos requisitos iniciais de construção da solução de integração é disponibilizar uma visão única dos conteúdos pelos sistemas consumidores, é necessário que a mesma seja capaz de unir as versões materializadas e virtualizadas do conteúdo de uma fonte de dados no momento em que há uma solicitação pelo sistema consumidor. Além disso, os conteúdos a serem materializados ou virtualizados precisam ser manipulados (aplicação de regras e transformações) de tal forma a produzirem sempre os mesmos resultados para os sistemas consumidores.

Outro ponto a ser observado é que as tecnologias usadas para realizar a integração devem implementar os processos vistos no Capítulo 2: extração, validação, transformação e carga. Na abordagem de virtualização, estes processos ocorrem em tempo de execução das consultas às fontes de dados, não sendo necessários os repositórios temporários de dados (ou *staging areas*). Caso não seja possível, o conteúdo da fonte de dados passa a ser integrado pela abordagem de materialização. Nesta abordagem, cada um dos processos pode ter um repositório associado e o conteúdo repousa lá até ser processado pelo próximo estágio na linha de integração, como mostra a Figura 2.2. Tradicionalmente, estes repositórios intermediários guardam os conteúdos de cada fase do processo em arquivos brutos, organizados em diretórios, ou em SGBDs relacionais. Contudo, estas duas alternativas geralmente limitam a manipulação e o gerenciamento do conteúdo. Enquanto que a guarda do conteúdo em arquivos limita o uso das potencialidades encontradas nos sistemas gerenciadores de banco de dados, os tradicionais bancos relacionais podem limitar a ingestão de conteúdos que possuam esquemas diferentes (quando existentes) ao relacional, transformando-os em meros atributos ininteligíveis aos processos comuns de consulta.

Porém, com o advento dos bancos NoSQL (Seção 2.1.5), surgiu a possibilidade do uso de repositórios não tradicionais para solucionar as limitações de ingestão que os SGBDs relacionais podem impor. Colocando de outra forma, com estas novas possibilidades de guarda do conteúdo, o banco de dados se adequa ao conteúdo e não ao contrário. Em um ambiente *Big Data*, a possibilidade de trabalhar com diferentes repositórios, especializa-

dos em determinadas tarefas, torna a ingestão de vários conteúdos, descritos de diversas formas, mais simples e eficiente. Neste sentido, o repositório dedicado ao processo de extração pode se beneficiar desta flexibilidade que muitos SGBDs NoSQL possuem, que é a possibilidade de carga sem a necessidade de descrição prévia do esquema do conteúdo (*schema-free*). Isto não só facilita a ingestão de qualquer conteúdo oferecido à solução de integração, como também a sua validação e manipulação.

O último ponto a ser observado é a escalabilidade dos repositórios temporários no processo de integração. Em um ambiente *Big Data*, espera-se que o volume dos conteúdos seja tal que as abordagens tradicionais não sejam suficientes para lidar com esse aspecto. Apesar da solução de integração não ter como característica recorrente a guarda dos conteúdos por um longo prazo, como em *datawarehouses*, não é possível garantir que isto não ocorrerá nos atuais ambientes de integração de dados. Além disso, a escalabilidade deve ser não só pensada em espaço, mas também em termos de processamento. É possível que uma manipulação de conteúdo se beneficie do processamento em múltiplos nós, como visto em soluções de processamento paralelo e distribuído. Sendo assim, os repositórios, especialmente aqueles ligados à transformação e à carga final, podem ser construídos levando em consideração a possibilidade de expansão tanto do volume quanto do processamento.

4.2 PROJETO DA SOLUÇÃO DE INTEGRAÇÃO

A FlexDI (*Flexible Data Integration*) é uma proposta de arquitetura de uma solução de integração capaz de, dinamicamente, alternar as abordagens de integração de cada fonte de dados, a partir de suas próprias características e das do ambiente de integração em que estão inseridas, minimizando não só a intervenção do administrador da solução, mas também o transporte de dados nas redes de comunicação. A Figura 4.1 mostra a arquitetura idealizada, constituída por quatro módulos: materialização, virtualização, disponibilização de dados e controle.

A apresentação em módulos é utilizada, neste momento, apenas para auxiliar na extração das funcionalidades que a solução de integração precisa exibir. A efetiva construção dos módulos se dará naturalmente a partir do projeto da solução e das decisões tecnológicas de implementação escolhidas. A próxima subseção apresenta os requisitos de cada módulo e os artefatos relevantes da UML para descrever o projeto preliminar, enquanto que as decisões de implementação serão apresentadas em subseção posterior. A documentação de todos os de casos de uso e do diagrama de classe preliminar da solução de integração encontra-se no apêndice deste trabalho.

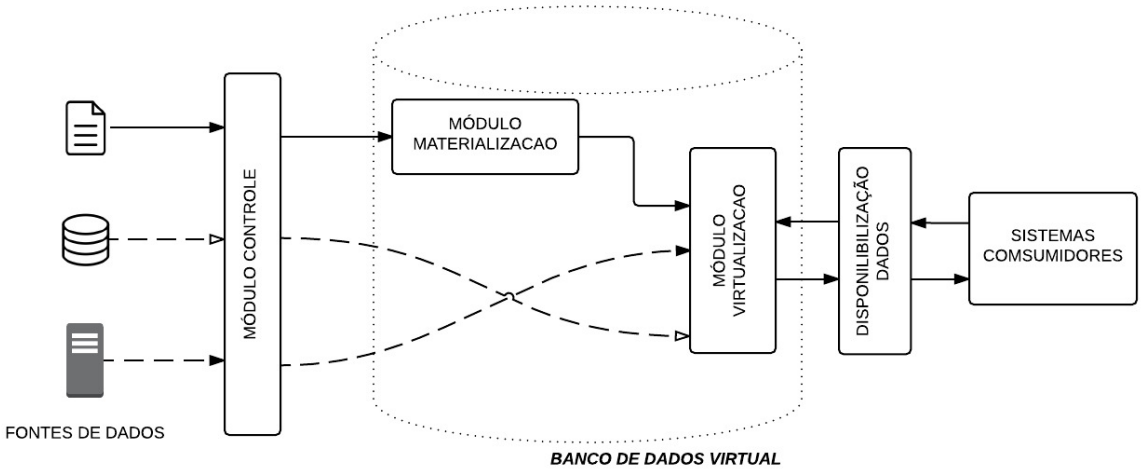

FIG. 4.1: Arquitetura Idealizada da Solução de Integração - FlexDI

4.2.1 MÓDULO DE MATERIALIZAÇÃO

O módulo de materialização é o elemento responsável pela internalização do conteúdo quando não é possível sua virtualização. Ele precisa ser capaz de executar os quatros processos tradicionais de integração de dados (extração, validação, transformação e carga), podendo ser apoiado, neste caso, por repositórios de dados temporários ao longo do processo. A entrega de um novo conteúdo produzido por uma fonte de dados pode ser feita de duas formas: ativa ou passivamente. Na primeira forma, a própria fonte de dados identifica que um novo conteúdo foi gerado e o envia na direção da solução de integração, enquanto que na segunda forma, a solução investiga periodicamente se um novo conteúdo foi produzido. Para os dois casos, há um repositório onde o conteúdo bruto repousa até a próxima fase do processo de integração. Uma vez identificado que um conteúdo está internalizado neste repositório, o módulo aplica as manipulações necessárias (regras e transformações), colocando aqueles bem sucedidos em um repositório de integração. Este será o repositório onde os sistemas consumidores acessarão os conteúdos já adequados para sua utilização.

Os casos de uso *Extrair Novo Conteúdo*,*Receber Novo Conteúdo*,*Adequar Novo Conteúdo* e *Remover Conteúdo Materializado* foram criados para descrever as funcionalidades necessárias deste módulo. Os dois primeiros casos se detêm na função de coleta do conteúdo da fonte de dados. O primeiro lida com as fontes de dados passivas, onde é necessário investigá-las periodicamente para determinar se um novo conteúdo foi gerado. Esta periodicidade de investigação expõe a necessidade de configuração de uma característica já

levantada anteriormente, que é a frequência de verificação de novo conteúdo pela solução de integração (f_V^{SI}). Já o segundo descreve a necessidade da solução de integração estar preparada para receber um novo conteúdo enviado por uma fonte de dados dentro de um canal comum de mensagens (*publish and subscribe*). Nesta condição, a fonte de dados tem um comportamento ativo, o que inviabiliza sua virtualização. O terceiro caso descreve a tradicional ação de manipular o conteúdo segundo as regras e transformações associadas a ele e internalizar o resultado bem sucedido em um repositório para posterior ingestão de um sistema consumidor. A última descrição se detém na ação de manutenção dos repositórios de extração e final de integração deste módulo, uma vez que estes repositórios não possuem espaços de armazenamento infinitos, refletindo assim outro aspecto importante apresentado no capítulo anterior, que é o tempo de vida do conteúdo na solução de integração (t_v^{SI}) e que pode influenciar na frequência de verificação dos sistemas consumidores (f_V^{SC}) em determinadas condições. O diagrama de caso de uso relativo a estas descrições e o diagrama de classes preliminar são mostrados na Figura 4.2.

4.2.2 MÓDULOS DE VIRTUALIZAÇÃO E DISPONIBILIZAÇÃO DE DADOS

O objetivo do módulo de virtualização é resgatar o conteúdo de uma fonte de dados da mesma forma que no módulo de materialização, porém em tempo de execução e somente quando o sistema consumidor assim o solicitar. Sendo assim, a virtualização pressupõe a inexistência de repositórios intermediários e a necessidade de consulta direta às fontes de dados. Esta última característica implica que somente as fontes de dados de comportamento passivo podem ser virtualizadas, excluindo a possibilidade de lidar com aquelas de comportamento ativo. Já a função do módulo de disponibilização de dados é entregar os conteúdos integrados das fontes de dados aos sistemas consumidores de maneira única, padronizada e transparente em relação à abordagem de integração sendo utilizada por cada fonte de dados participante do ambiente de integração. Ou seja, o módulo precisa ser capaz de entregar o conteúdo de uma fonte de dados em um formato de troca conhecido e de, no momento da solicitação, unir suas versões materializadas e virtualizadas em tempo de execução.

Uma vez que o módulo de virtualização precisa processar o conteúdo de uma fonte de dados da mesma forma que o módulo de materialização o faria, seria desejável que o mesmo pudesse emular, de alguma forma, um repositório de integração onde os conteúdos nele processados pudessem ser consumidos. Ainda mais interessante seria se o módulo pudesse emular um sistema gerenciador de banco de dados. Este SGBD virtual então

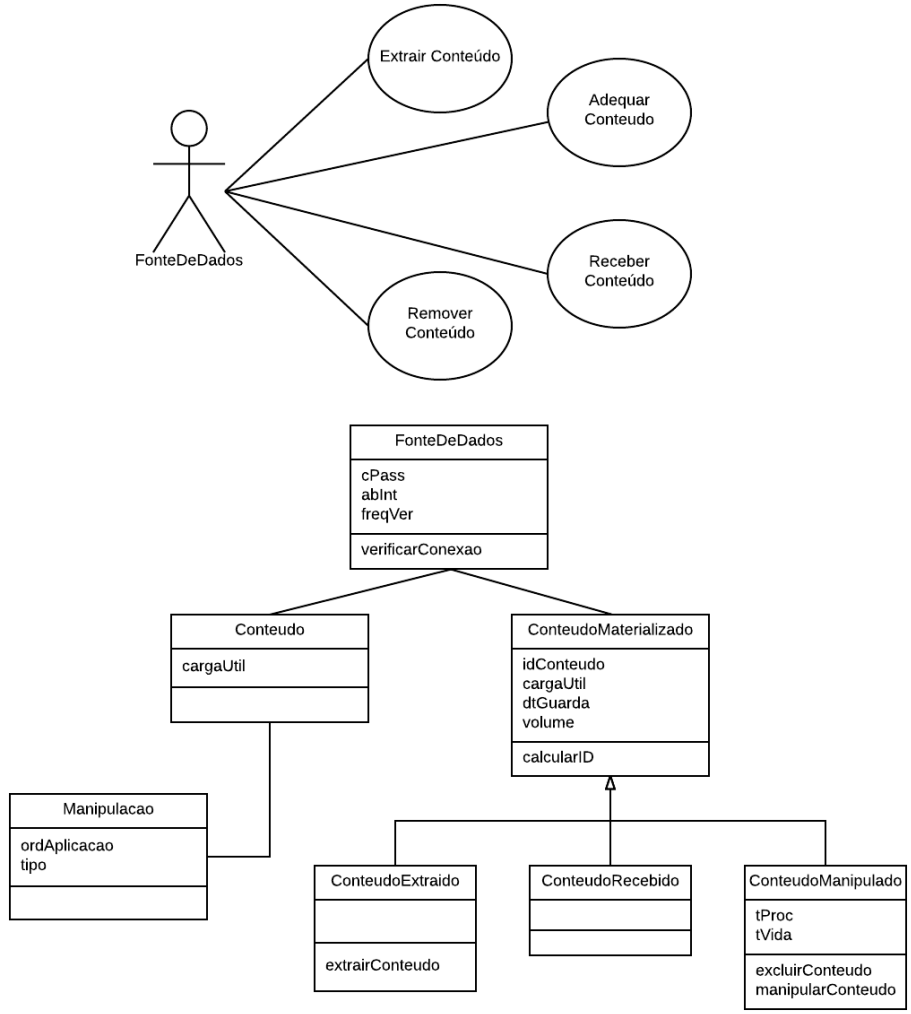

FIG. 4.2: Artefatos UML - Projeto Preliminar - Materialização

trataria as diversas fontes de dados, cada uma, como um conjunto de *tabelas* de um banco de dados, e o acesso ao conteúdo se daria por meio de consultas como as encontradas em diversos outros sistemas de gerenciamento. Como efeito secundário deste entendimento, ao admitir a possibilidade de criação de um módulo de virtualização que emule as funções de um SGBD, a questão de unir conteúdos tomaria contornos menos complexos.

Outro requisito a ser considerado neste módulo é a sua capacidade de conectar-se com as mais diversas fontes de dados. Ao considerar que o repositório de integração do módulo de materialização também é mais uma fonte de dados para módulo de virtualização, um SGBD virtual seria capaz de recuperar e unir as versões materializadas e virtualizadas de um conteúdo. Consequentemente, transferir-se-ia a responsabilidade da união dos conteúdos do módulo de disponibilização de dados para o módulo de virtualização. A Figura 4.3 mostra esta percepção.

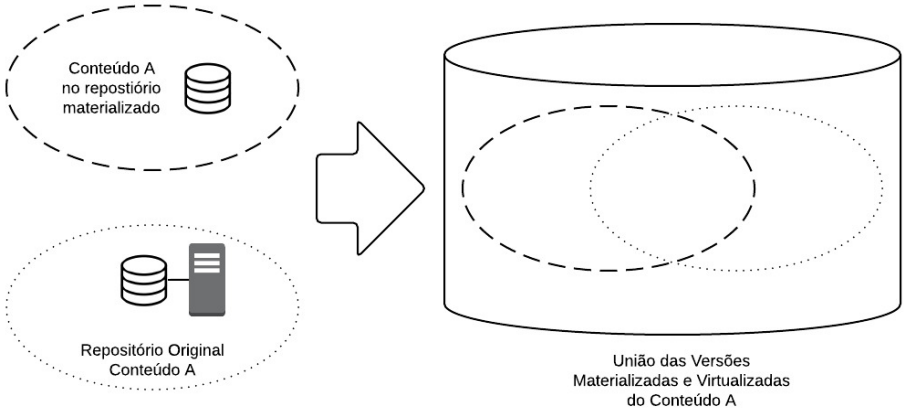

FIG. 4.3: União dos Repositórios Físico e Virtual

Assim, considerando-se como factível a construção deste tipo de SGBD, quando o módulo de disponibilização de dados procurar pelas versões de um conteúdo, será necessário apenas questionar o banco de dados virtual do módulo de virtualização. E assim, sua responsabilidade será apenas a de estruturar o conteúdo solicitado pelo sistema consumidor em um formato conhecido. Há de se ressaltar que a união de conteúdos remete a uma funcionalidade tradicional dos sistemas gerenciadores de bancos de dados relacionais, porém não é exclusividade dos mesmos.

Em uma primeira abordagem, devido à separação lógica dos módulos, foram criados dois casos de uso: *Disponibilizar Conteúdo* e *Virtualizar Conteúdo*. Porém, na elaboração da descrição, notou-se que o caso *Virtualizar Conteúdo* era desnecessário, uma vez que a solução de integração deve se comportar, para estes casos, apenas como um elemento de ligação, sendo praticamente transparente para a solicitação dos sistemas consumidores. Logo, apenas o caso de uso *Disponibilizar Conteúdo* foi desenvolvido. A Figura 4.4 mostra os diagramas de caso de uso e de classe preliminar para esta funcionalidade.

4.2.3 MÓDULO DE CONTROLE

O módulo de controle é o responsável pela execução do objetivo principal deste trabalho, que é a seleção da abordagem de integração mais adequada para uma fonte de dados a partir das suas características e das do ambiente de integração. Contudo, para que isso seja possível, há outras atividades que precisam ser exercidas pelo módulo, tais como as de monitoração das características e de manutenção de cadastros. Como a totalidade das funcionalidades está descrita no apêndice deste trabalho, somente as funcionalidades mais relevantes ao desenvolvimento do projeto serão descritas nas próximas subseções.

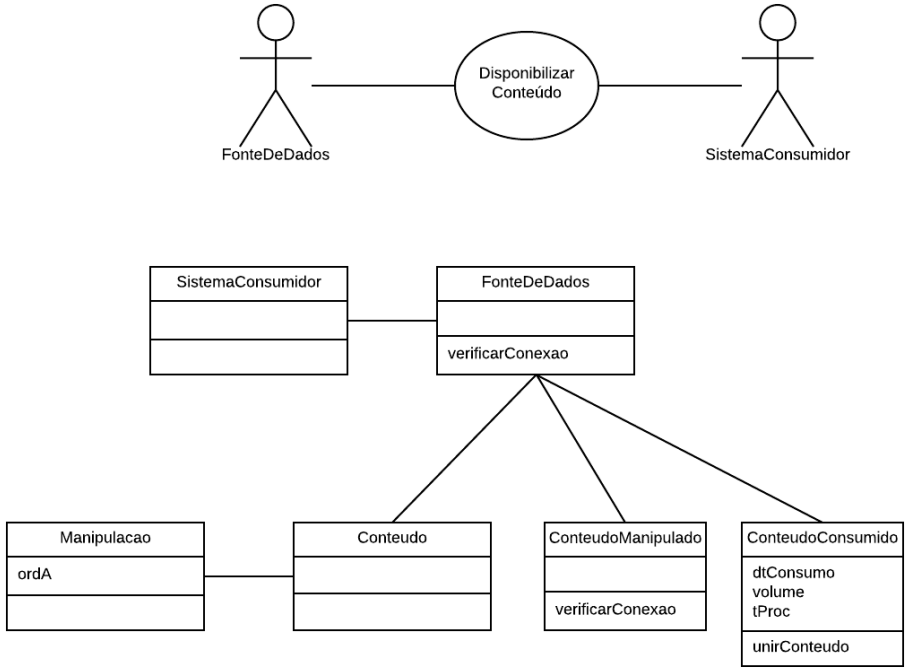

FIG. 4.4: Artefatos UML - Projeto Preliminar - Disponibilização de Conteúdo e Virtualização

4.2.3.1 MANIPULAÇÃO DO CONTEÚDO

Como colocado na Seção 4.2, as manipulações estão relacionadas ao conteúdo e independem da abordagem de integração para serem aplicadas. Dessa forma, elas precisariam ser compartilhadas entre os módulos de materialização e virtualização. Dada a possibilidade de implementação destes módulos por meio de tecnologias diferentes, seria necessário pensar em formas de tradução destas manipulações de tal sorte a serem utilizáveis por ambos os módulos. Contudo, a utilização do repositório de integração do módulo de materialização como uma nova fonte de dados pelo módulo de virtualização permitiu um simplificação neste contexto, uma vez que as manipulações de conteúdo comuns para ambas as abordagens podem ser simplesmente aplicadas no módulo de virtualização. A aplicação de uma manipulação no módulo de materialização só é realizada nos casos onde o módulo de virtualização é incapaz de executá-la.

Para descrever esta funcionalidade, foi criado o caso de uso *Associar Manipulações*. Nele, as manipulações são associadas ao conteúdo de cada fonte de dados, por ordem de aplicação, e indicado se é necessária sua execução quando a abordagem de integração sendo utilizada for a de materialização. Os artefatos da UML que tratam desta funcionalidade são mostrados na Figura 4.5.

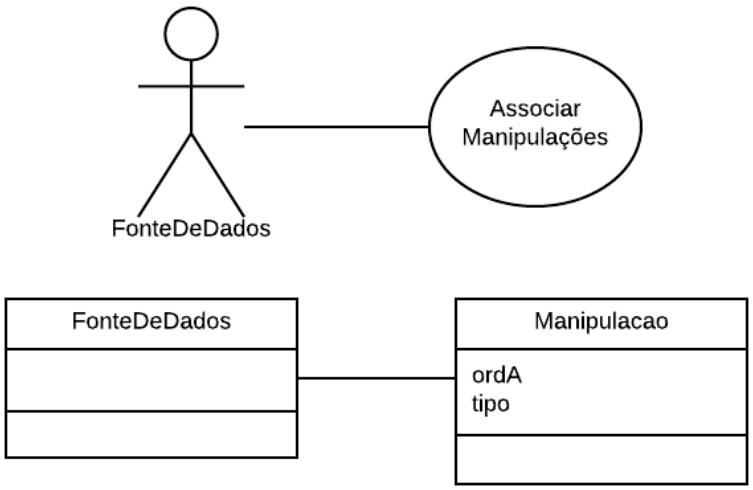

FIG. 4.5: Artefatos UML - Projeto Preliminar - Manipulação de Conteúdos

TAB. 4.1: Aspectos Estáticos

Atributo	Descrição	Domínio	Entidade
cPass	Comportamento Passivo do Invólucro	Lógico{True,False}	Fonte de Dados
cResp	Capacidade de Resposta do Invólucro	Lógico{True,False}	Fonte de Dados
eStxe	Existência de Sintaxe no Conteúdo	Lógico{True,False}	Fonte de Dados
eMlog	Existência Modelo de Lógico no Conteúdo	Lógico{True,False}	Fonte de Dados

4.2.3.2 MONITORAÇÃO DAS CARACTERÍSTICAS

Antes de selecionar a abordagem de integração mais adequada para uma determinada fonte de dados, é necessário monitorar periodicamente as características das fontes de dados e do ambiente de integração relevantes para tal ação, como demonstrado no capítulo anterior. As tabelas 4.1e 4.2 trazem mais uma vez essas características.

Por hipótese, os aspectos classificados como estáticos não mudam ao longo do tempo. A mudança destes aspectos somente se dá na inserção e na atualização de uma fonte de

TAB. 4.2: Aspectos Dinâmicos

Atributo	Descrição	Domínio	Entidade
t_v	Tempo de Vida do Conteúdo	Decimal	Fonte de Dados
f_a	Freq. de Atualização do Conteúdo	Inteiro	Fonte de Dados
f_V	Freq. de Verificação do Conteúdo	Inteiro	Fonte de Dados, Sistema Consumidor
t_T	Tempo de Transporte entre Entes	Decimal	Enlaces
v_c	Volume do Conteúdo	Decimal	Fonte de Dados

dados participante do ambiente de integração. Há duas formas de analisar cada um dos aspectos estáticos: uma manual, por meio de questionário ao administrador do sistema, ou automático. Alguns projetos, como o projeto Apache Tika (FOUNDATION, 2016), propõem uma análise automática dos metadados de um conteúdo de uma fonte de dados, o que poderia permitir a análise e a seleção automática de alguns aspectos estáticos como a existência de sintaxe e o modelo lógico. Contudo, sua implementação foge ao escopo de conteúdo e tempo destinado para este trabalho e, sendo assim, o ajuste destas características na solução de integração fica a cargo do administrador do sistema. Como não há necessidade de realizar uma monitoração contínua destes aspectos, basta a investigação dos valores atuais destas características no momento da seleção da abordagem. Para facilitar a análise da abordagem de integração, os aspectos estáticos são construídos como atributos lógicos, pois é possível rapidamente verificar se a fonte de dados é passível ou não de virtualização apenas aplicando uma função lógica E a este conjunto de características.

Diferente dos aspectos estáticos, os dinâmicos precisam de uma monitoração frequente para que seja possível uma avaliação contínua da abordagem de integração sendo utilizada por cada fonte de dados. A monitoração destas características pode ser realizada de duas maneiras: criando funcionalidades para tal fim ou aproveitando a ação de outras funcionalidades do sistema para realizar tal medição. A primeira produz casos de uso mais simples, porém aumenta o número de casos para serem gerenciados e implementados. Já a segunda alternativa minimiza a criação de novas funcionalidades, apesar de aumentar a complexidade das existentes. A abordagem utilizada em tempo de projeto foi analisar cada um dos atributos e verificar se a oportunidade de medir certo aspecto dentro de outra funcionalidade era mais apropriado do que criar um caso de uso somente para tal ação.

O tempo de vida(t_v^{FD}) e a frequência de atualização(f_a^{FD}) do conteúdo de uma fonte de dados com comportamento passivo podem ser medidos toda vez que a extração é acionada. No momento da extração, é possível guardar a data e a hora em que determinado conteúdo foi encontrado e, consequentemente, estimar o valor destas duas características. Considerando o exposto, a medida de tempo de vida e a frequência de atualização do conteúdo na origem pode ser realizada dentro das funcionalidades *Extrair Conteúdo* e *Receber Conteúdo*. Já a avaliação do tempo de vida do conteúdo materializado(t_v^{SI}) poderia ser definida em função do volume médio do conteúdo e do espaço disponível no repositório destinado para tal função. Contudo, a descoberta desta relação está fora do escopo deste trabalho, sendo, neste caso, definida pelo administrador do sistema no momento do cadastro da fonte de dados.

A frequência de atualização no repositório de integração(f_a^{SI}) pode ser estimada utilizando o tempo necessário para transportar o conteúdo original para o repositório de extração da solução de integração(t_T^{FDSI}) e o tempo de processamento(t_p) para adequá-lo e salvá-lo no repositório de integração, sendo então possível medir este último juntamente com o processo de adequação descrito no caso de uso *Adequar Conteúdo*. Como colocado no capítulo anterior, é factível a medição do tempo de transporte de um conteúdo entre a fonte de dados e a solução de integração, uma vez que esta ação está no escopo de administração da própria solução. Contudo, o mesmo não pode ser esperado do sistema consumidor. Para transpor esta limitação, o tempo de transporte entre os entes do ambiente de integração é estimado pelo volume de conteúdo transportado(v_c) e pela latência do enlace de comunicação(t_l) que liga os entes do ambiente. Diferente do volume, que pode ser propriamente medido no momento da extração de um conteúdo, a medição de latência do enlace não se enquadra no escopo de funcionalidades descritas até o momento.

A frequência de verificação de novo conteúdo de uma fonte de dados(f_V^{SI}) é calculada a partir do seu tempo de vida e da sua frequência de atualização, do tempo de transporte do conteúdo entre a fonte de dados e o sistema e da probabilidade de perda admitida pelo sistema($p(x = 0)^{SI}$). Assim, sua monitoração não é necessária. Por outro lado, como não é possível ajustar a frequência de verificação do sistema consumidor(f_V^{SI}), devido a restrições de escopo de administração, faz-se necessário sua monitoração, uma vez que faz parte da avaliação da abordagem de integração mais adequada da fonte de dados sendo consumida. Essa medição pode ser realizada toda vez que houver a solicitação de consumo de uma determinada fonte de dados, sendo então incluída no caso de uso *Disponibilizar Conteúdo*.

A medição do tempo de transporte de conteúdo entre os entes do ambiente de integração não se encaixa como um funcionalidade clara e pertinente nas descrições de caso de uso discutidas até o momento. Portanto, faz-se necessária a criação de funcionalidade específica para tal ação. Segundo a arquitetura idealizada no início desta seção, há três enlaces principais: Fonte de Dados - Solução de Integração (FDSI), Solução de Integração - Sistema Consumidor (SISC) e Fonte de Dados - Sistema Consumidor (FDSC). O transporte do conteúdo no enlace FDSI(t_T^{FDSI}) pode ser estimado pelo volume do conteúdo extraído(v_c^{FD}) e a latência de rede do enlace(t_l^{FDSI}). Da mesma forma, o transporte do conteúdo no enlace SISC(t_T^{SISC}) pode ser estimado pelo volume do conteúdo integrado(v_c^{SI}) e a latência de rede do enlace(t_l^{SISC}). Porém, diferente das duas medições anteriores, a avaliação do tempo de transporte no enlace FDSC(t_T^{FDSC}) mostra-se inexequível, pois os dois entes estão, a princípio, fora do escopo de administração do ambiente de integração,

uma vez que a medição de latência deveria ser feita por um destes entes. Para superar este contratempo, decidiu-se por estimar este tempo de transporte como a soma dos tempos de transporte dos enlaces FDSI e SISC e o tempo de processamento(t_p) destinado a adequar o conteúdo para o sistema consumidor.

Nota-se com o exposto até o momento nesta subseção que apenas a medição do tempo de transporte do conteúdo entre os entes do ambiente de integração necessita de uma descrição de caso de uso específica. Todas as outras características são medidas aproveitando a ação de outras funcionalidades do sistema sendo projetado. Logo, foi criado o caso *Monitorar Enlace* para descrever a funcionalidade necessária para medir o tempo de transporte do conteúdo entre os entes do ambiente de integração, sendo os diagramas de caso de uso e de classe representados na Figura 4.6.

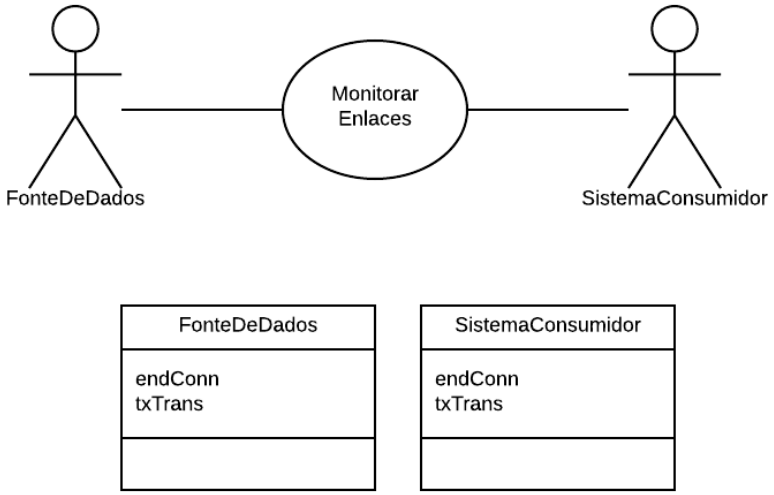

FIG. 4.6: Artefatos UML - Projeto Preliminar - Monitoração de Enlaces

4.2.3.3 SELEÇÃO DA ABORDAGEM DE INTEGRAÇÃO

A seleção da abordagem de integração mais apropriada para uma determinada fonte de dados é o núcleo do desenvolvimento deste trabalho. Nesta funcionalidade, o sistema necessita recuperar as características das fontes de dados e do ambiente de integração ao final de cada materialização ou no término da solicitação de conteúdo por um sistema consumidor. A partir deste momento, o fluxo de decisão mostrado na Figura **??** é aplicado, selecionando assim a abordagem de integração mais adequada para cada fonte de dados naquele instante do tempo de vida do ambiente.

Primeiro, os aspectos estáticos (comportamento passivo, capacidade de resposta, sin-

taxe e modelo lógico) são recuperados do cadastro da fonte de dados uma vez que qualquer negativa neste ponto frustra a possibilidade de virtualização da fonte de dados. Para facilitar esta avaliação, os aspectos estáticos são definidos como atributos lógicos, sendo necessário apenas a execução de um simples E lógico para determinar se há ou não a possibilidade de virtualização da fonte de dados sendo avaliada. Caso todos os aspectos estáticos habilitem a possibilidade de virtualização, os aspectos dinâmicos são então avaliados segundo as formulações apresentadas na Seção 3.1.4 do Capítulo 3.

Na recuperação dos aspectos dinâmicos, é necessário estabelecer um valor que representará as medições feitas no momento de aplicação do fluxo decisório. A área da estatística descritiva provê vários métodos para caracterizar amostras por meio de funções como a média ou a mediana, assim como a avaliação de valores suspeitos de não pertencer a tal conjunto de dados (análise interquartílica)(ZENTGRAF, 2001). Além disso, é esperado que alteração da abordagem de integração se dê de forma suave, evitando assim mudanças abruptas ou desnecessárias em um determinado intervalo de tempo. Há vários métodos que podem ser implantados, desde simples análises baseadas na média e desvio padrão até análises mais sofisticadas utilizando algoritmos de aprendizado de máquina. Sendo assim, fica a cargo da implementação utilizar o método de representação mais conveniente das características dinâmicas para aplicá-las no fluxo decisório apresentado. Outro ponto importante a ser mencionado é que, mesmo que a virtualização não seja possível, é desejável que as frequências de verificação de novos conteúdos do módulo de materialização(f_V^{SI}) e dos sistemas consumidores(f_V^{SC}) sejam ajustados ou avaliados a partir dos valores calculados. Espera-se que, com este ajuste, o consumo de processamento e memória no sistema seja reduzido.

Dado o exposto, foram criados dois casos de uso: *Avaliar Abordagem* e *Ajustar Materialização*. O primeiro lida com a avaliação propriamente dita da abordagem de integração, enquanto que o segundo segrega a funcionalidade de ajuste da frequência de verificação de novos conteúdos pelo sistema. A separação das funcionalidades em dois casos de uso diferentes deve-se ao fato que o segundo caso não só estende as funcionalidades do primeiro como também do caso *Adequar Conteúdo*. Os artefatos da UML que representam tais funcionalidades são mostrados na Figura 4.7

4.3 IMPLEMENTAÇÃO DA SOLUÇÃO DE INTEGRAÇÃO

A implementação do projeto da solução de integração teve como foco a rápida construção para que fosse possível realizar os testes para confirmar ou refutar as hipóteses

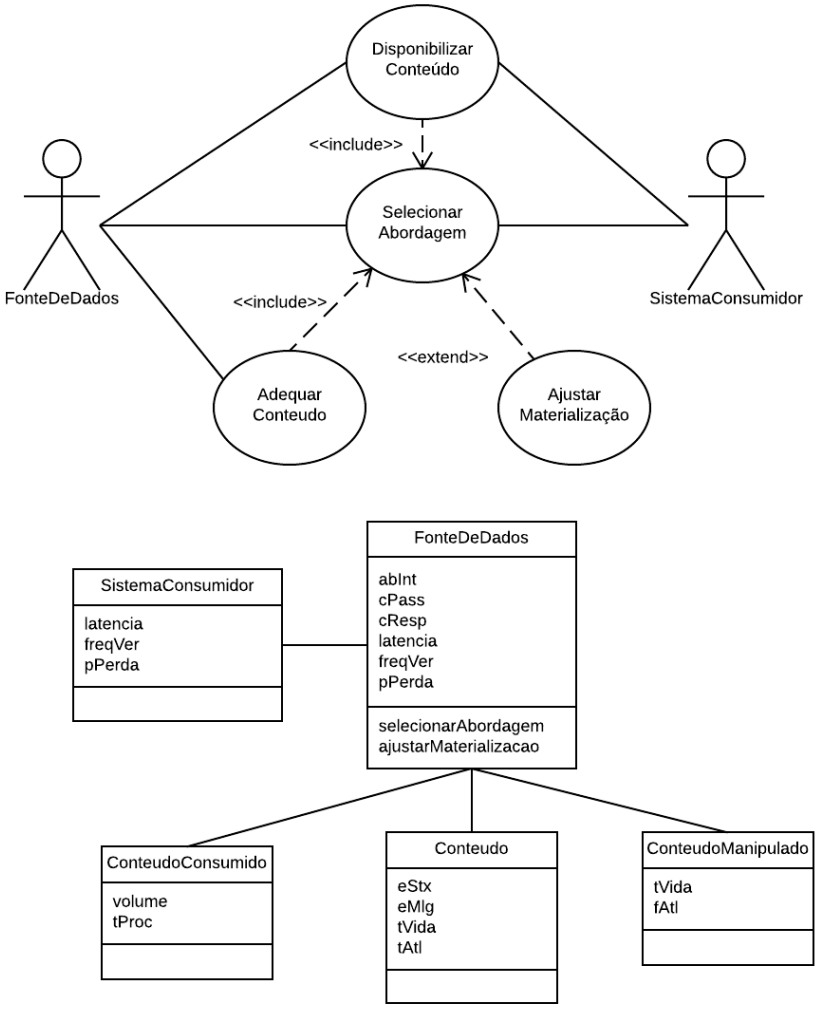

FIG. 4.7: Artefatos UML - Projeto Preliminar - Seleção da Abordagem de Integração

levantadas no início deste trabalho dentro do escopo de tempo estabelecido. Para tanto, foram pesquisadas soluções que pudessem suportar os requisitos de cada um dos módulos apresentados anteriormente e somente foram criados artefatos computacionais onde não havia alternativa disponível. As próximas subseções apresentam as escolhas mais relevantes utilizadas, deixando outros detalhes mais simples de implementação para consulta no apêndice.

4.3.1 TEIID VIRTUALIZATION SERVER

Um dos desafios postos no projeto da solução de integração foi a criação do módulo de virtualização, em especial do banco de dados virtual, onde as fontes de dados seriam tratadas como meras "tabelas"pelo sistema, sem a necessidade de transporte de seus conteúdos antes da efetiva solicitação por um sistema consumidor. Felizmente, a pesquisa

por soluções existentes apontou para um sistema de código aberto que não só satisfaz as necessidades da virtualização como também da disponibilização de dados. Este sistema é chamado de Teiid (RED HAT, 2016a).

Segundo seus desenvolvedores, o Teiid é um sistema de virtualização de dados que permite que aplicações possam usar dados de múltiplos repositórios heterogêneos. Construído em Java e suportado pelo servidor de aplicação JBoss, os dados são acessados e integrados em tempo de execução, por meio de abstração e federação, sobre vários repositórios distribuídos sem copiar ou mover previamente os dados de sua origem. O sistema possui quatro blocos construtivos principais: os modelos, os tradutores e adaptadores, os conectores e os bancos de dados virtuais propriamente ditos. A Figura 4.8 mostra o conceito por trás do sistema.

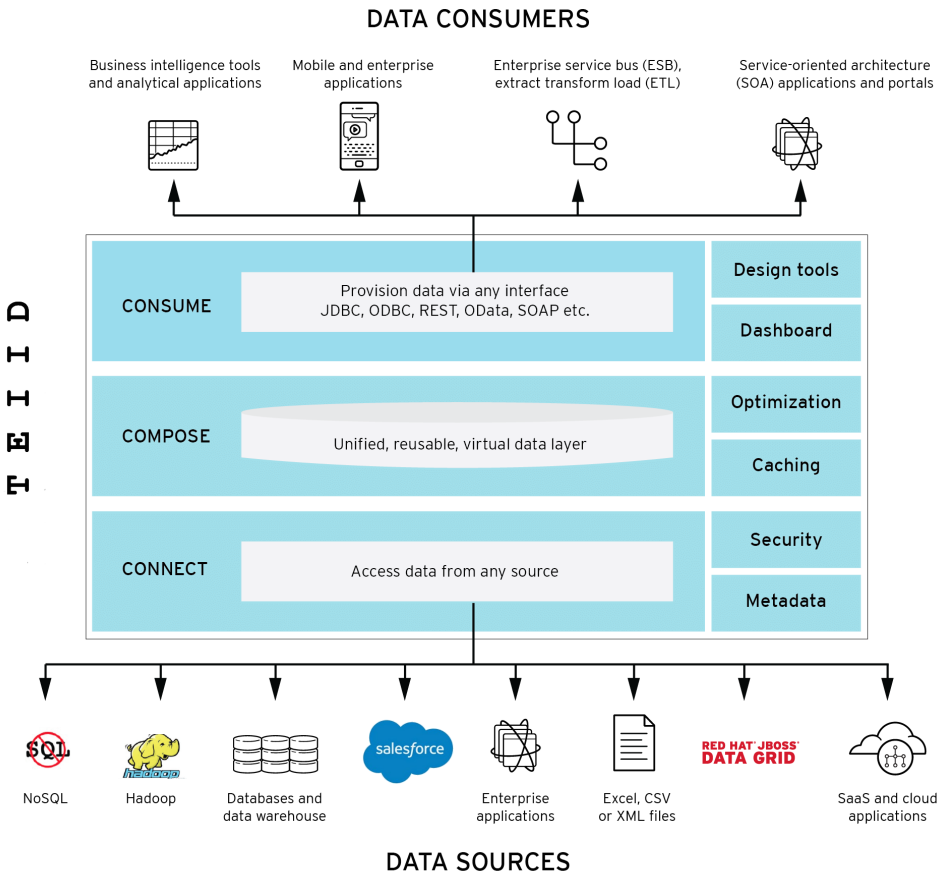

FIG. 4.8: Modelo Conceitual Teiid (RED HAT, 2016a)

Como qualquer outro modelo, os modelos do Teiid representam um conjunto de construtos de informação e possuem dois tipos: os de fonte e os de visão. Os modelos de fonte definem as estruturas e as características dos conteúdos das fontes de dados. O sistema usa esta informação para acessar o conteúdo das múltiplas fontes de dados, provendo uma

interface única para os sistemas consumidores. Em adição aos modelos de fonte, o Teiid também provê a habilidade de definir uma variedade de modelos de visão. Eles podem ser usados como um nível de abstração acima da camada física, sendo que o conteúdo apresentado para os sistemas consumidores pode ser expresso em termos de regras de negócio, utilizando-se de manipulações dos modelos de fonte e de visão previamente criados para prover tal apresentação.

Os tradutores e os adaptadores são a ponte entre os modelos e as fontes de dados físicas. Os tradutores provêm uma camada de abstração entre o motor de consulta do Teiid e a fonte de dados propriamente dita, convertendo as consultas colocadas para o sistema em comandos específicos da fonte e executando-as juntamente com um adaptador. Da mesma maneira, os tradutores têm a capacidade de converter o resultado proveniente das fontes físicas em um formato que o motor de consulta do Teiid espera. Já os adaptadores provêm conectividade com as fontes de dados físicas, atuando em conjunto com os tradutores para disparar comandos nativos e recuperar seus resultados. Já os conectores são reponsáveis por prover o acesso à informação federada através de classes e interfaces JDBC (SQL ou XQuery), ODBC (SQL ou XQuery) e SOAP (Web Services)

O banco de dados virtual(VDB) é um *container* dos componentes usados para integrar as fontes de dados: os modelos, os tradutores e adaptadores e os conectores. Ele pode conter um ou mais modelos representando o conteúdo a ser integrado e os expõe aos sistemas consumidores. É importante salientar que os modelos precisam estar em um estado válido para que o VDB possa ser utilizado. Enquanto que a validação de um único modelo significa que ele precisa estar em um estado completo e consistente (não há pedaços faltantes ou referências a entidades não existentes), a validação de múltiplos modelos verifica também se todas as interdependências estão presentes e são resolvíveis.

Há várias formas de construir um banco de dados virtual para operar no servidor Teiid. Uma delas é construí-lo utilizando uma ferramenta gráfica baseada na interface do Eclipse - o Teiid Designer (RED HAT, 2016b). Utilizando-o como uma nova perspectiva nesta IDE, é possível não só definir as fontes, as visões e as transformações que se fazem necessárias, como também testes para resgatar o conteúdo federado por meio de uma interface SQL. Uma vez que o banco de dados virtual está montado e testado na ferramenta, basta sua implementação no servidor Teiid para poder oferecer aos sistemas consumidores um acesso único às fonte de dados participantes do ambiente de integração. A Figura 4.9 mostra um exemplo da interface de construção do banco de dados virtual.

O uso do servidor Teiid juntamente com o Teiid Designer para a criação do banco de dados virtual possibilitou uma rápida implementação das funcionalidades descritas no

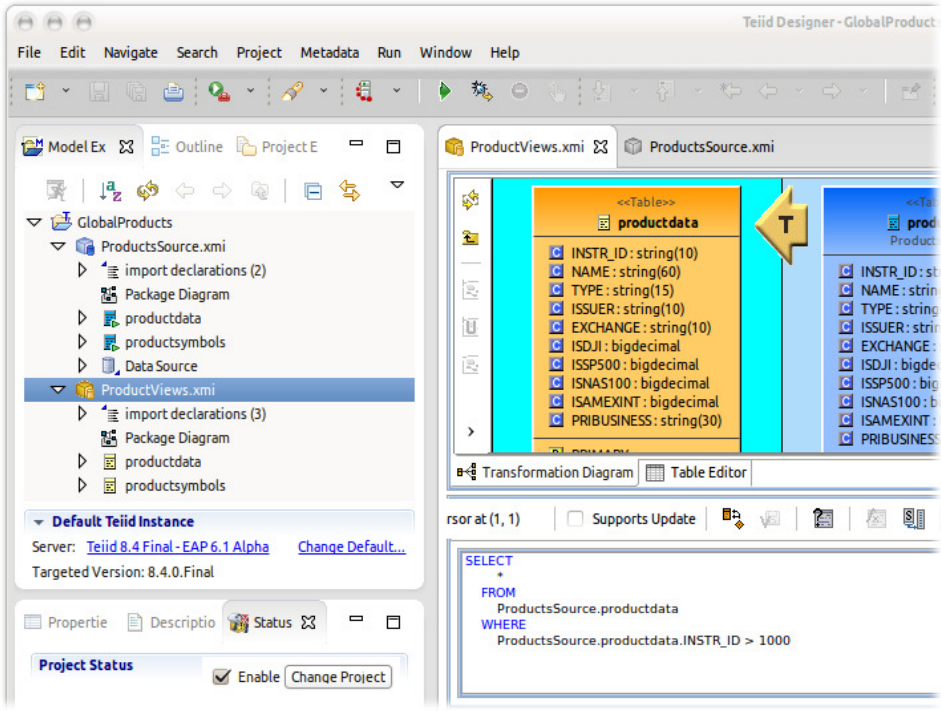

FIG. 4.9: Exemplo Teiid Designer(RED HAT, 2016b)

caso de uso *Disponibilizar Conteúdo*. Não só foi possível criar um banco de dados virtual por meio de uma interface gráfica, como também o uso imediato das funcionalidades de acesso ao conteúdo federado. Para prover rapidez no desenvolvimento e nos testes deste trabalho, foi utilizado o acesso via o componente JDBC oferecido junto com o pacote de instalação do servidor de virtualização. Além das funcionalidades descritas no caso de uso citado anteriormente, o servidor proveu também outras facilidades de monitoração e gerenciamento. A disponibilização de vários tipos de contexto para geração de logs de atividade permitiu a análise da frequência de verificação do sistema consumidor, caraterística esta fundamental para avaliação da abordagem de integração a ser utilizada em determinado instante do processo.

4.3.2 APACHE HBASE

Outro desafio encontrado foi a escolha apropriada dos repositórios de extração e de integração para conteúdos a serem materializados. Por se tratar de uma proposta de arquitetura em um ambiente *Big Data*, dois requisitos foram considerados para a escolha de tais repositórios: a sua escalabilidade e a possibilidade de ingestão de conteúdo sem prévia descrição de esquema. O primeiro requisito é encontrado tanto nos sistemas gerenciadores de bancos de dados relacionais quanto nos sistemas gerenciadores de bancos de dados

NoSQL. Contudo, a forma como cada tipo de SGBD é escalado é diferente.

Os SGBDs relacionais foram concebidos inicialmente para oferecer uma escalabilidade vertical. Porém, ao longo do tempo, estes sistemas começaram a proporcionar escalabilidade horizontal também. Mesmo assim, sua implementação em SGBDs relacionais é considerada mais complexa do que com SGBDs NoSQL (MICHAEL et al., 2007), que pode ser mais facilmente obtida simplesmente adicionando computadores de pequeno e médio porte ao *cluster*. Nesta linha, esta implementação optou pela utilização de SGBDs NoSQL para, quando necessário, escalar horizontalmente.

O segundo requisito é ligado principalmente ao repositório de extração. Até o momento, os tradicionais sistemas gerenciadores de bancos de dados relacionais não apresentam a mesma flexibilidade em relação à descrição prévia do esquema presente em vários sistemas NoSQL, o que pode dificultar a ingestão de certos conteúdos, principalmente aqueles com padrões e formatos recém criados. Sendo assim, a pesquisa concentrou-se no levantamento das características de SGBDs NoSQL. Nesta pesquisa, dois deles se sobressaíram para resolver os requisitos propostos: o Apache HBase e o Apache Accumulo. Estes dois sistemas *wide-column* possuem como características a flexibilidade na descrição do esquema e a escalabilidade dentro do ambiente Hadoop. No final, a decisão de implementação pendeu para o Apache HBase pelo simples fato que este repositório possui maior abrangência de conectores já prontos para uso, como encontrados, por exemplo, no Pentaho Data Integrator (PDI) e no servidor de virtualização Teiid. Com estas características, o HBase pode ser usado tanto como repositórios de extração quanto de integração.

O Apache HBase é uma implementação Java do BigTable da Google. Assim como o BigTable, o HBase é definido por seus desenvolvedores como um mapeamento multidimensional ordenado de dados esparsos, distribuídos e persistentes. A definição concisa dificulta um pouco o entendimento de seu propósito. Como colocado em (DEROOS; COSS, 2014), os atributos de distribuição e persistência estão relacionados a sua direta ligação ao sistema de arquivo HDFS do ambiente Hadoop . Este estreito relacionamento confere ao HBase a capacidade de escalabilidade horizontal ao qual o ambiente Hadoop se propõe. A Figura 4.10 mostra a arquitetura do ambiente Hadoop e como o HBase se situa nesta arquitetura.

A condição esparsa dos dados no HBase está ligada à flexibilidade no trato dos esquemas uma vez que é capaz de lidar com atributos nulos ou novos atributos de maneira mais eficiente que os sistemas gerenciadores que necessitam de descrição prévia de esquema. Por fim, o mapeamento multidimensional ordenado está ligado ao modelo de dados suportado pelo HBase. Segundo deRoos e Coss (2014), o repositório do HBase consiste em uma

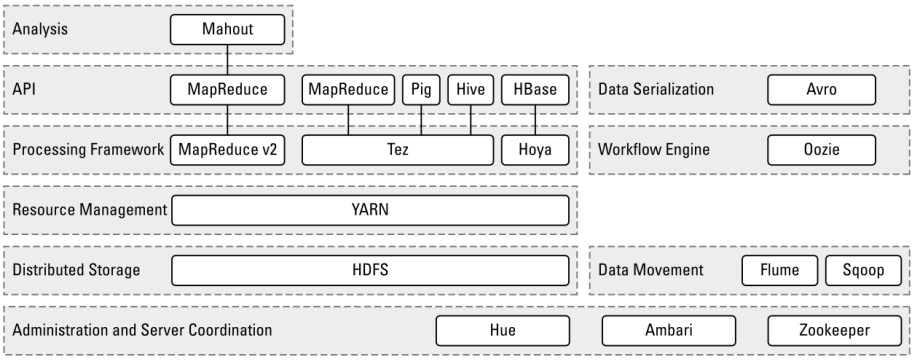

FIG. 4.10: Arquitetura Ambiente Hadoop(DEROOS; COSS, 2014)

ou mais tabelas com linhas indexadas por chaves, rotuladas temporalmente e compostas por colunas agrupadas em famílias, como mostra o exemplo da figura 4.11 . A lista abaixo detalha cada um dos elementos do modelo lógico de dados adotado pelo HBase:

Row Key	*Column Family:{Column Qualifier:Version:Value}*
00001	CustomerName: {'FN': 1383859182496:'John', 'LN': 1383859182858:'Smith', 'MN': 1383859183001:'Timothy', 'MN': 1383859182915:'T'} ContactInfo: {'EA': 1383859183030:'John.Smith@xyz.com', 'SA': 1383859183073:'1 Hadoop Lane, NY 11111'}
00002	CustomerName: {'FN': 1383859183103:'Jane', 'LN': 1383859183163:'Doe', ContactInfo: { 'SA': 1383859185577:'7 HBase Ave, CA 22222'}

FIG. 4.11: Exemplo Tabela HBase(DEROOS; COSS, 2014)

– **Chave de Linha (*Row Key*)**: As chaves são implementadas por conjunto de bytes e ordenadas em uma ordem lexicografica byte a byte, da esquerda para direita. Portanto, dada duas chaves, aquela que tiver o menor valor será a mais antiga armazenada. Apesar de óbvio para chaves numéricas, o mesmo não pode ser tido

para aquelas formadas por valores não numéricos. Como colocado em (DEROOS; COSS, 2014), chaves alfanuméricas são comuns no HBase e é necessário lembrar que este SGBD é desenvolvido em Java, que representa os caracteres pelo padrão Unicode ao invés do usual padrão ASCII. Essa diferença pode levar a problemas de desempenho na procura de dados devido a uma mero equívoco na interpretação de como a linguagem utiliza seu dicionário de caracteres.

- **Famílias de Colunas (*Column Family*)**: Ao criar uma tabela no HBase, é solicitado ao desenvolvedor a criação de uma ou mais famílias de coluna. Apesar de geralmente as famílias de colunas permanecerem fixas ao longo do tempo de vida da tabela no HBase, novas famílias podem ser adicionadas quando necessário. Há duas recomendações importantes dadas pelos desenvolvedores do HBase: a limitação no número de famílias e o armazenamento de dados com padrões de acesso similares. Na primeira, os desenvolvedores sugerem que as famílias sejam menores que 4. Já na segunda, a recomendação é agrupar dados com padrões de acesso similares na mesma família de colunas, uma vez que são agrupadas juntas em disco e esta condição reduz o acesso geral ao disco e aumenta o desempenho do SGBD.

- **Qualificadores de Coluna (*Column Qualifier*)**: Os qualificadores de coluna são nomes específicos designados aos valores dos dados para garantir a capacidade de recuperá-los quando solicitados. Diferente das famílias de coluna, os qualificadores podem ser virtualmente ilimitados em conteúdo, tamanho e número. Se um qualificador é omitido, o próprio sistema se encarrega de criá-lo. A princípio, qualquer tipo e quantidade de bytes pode ser usado para se criar um qualificador. Porém, como o mesmo faz parte da chave de recuperação de um determinado valor, não é recomendável utilizar nomes extensos para qualificadores. Neste caso, a recomendação é utilizar abreviaturas para designá-los.

- **Versão (*Version*)**: O número entre o qualificador e o valor representa a versão, o rótulo temporal que cada valor tem na tabela. Esta característica permite a rápida identificação de diferentes versões de um determinado qualificador, sendo que, por padrão, o HBase sempre retorna o valor mais atual caso não haja especificação em outro sentido. A versão representa a quantidade de milissegundos passados desde a meia-noite de primeiro de janeiro de 1970 (UTC).

- **Valor (*Value*)**: É efetivamente o resultado solicitado dada uma chave de procura.

Apesar do HBase ser classificado como um repositório *wide-column*, a busca por valores é baseado no paradigma da recuperação do par chave-valor (*key-value store*). Neste caso, a chave de procura é composta pela chave de linha, família de coluna, qualificador de coluna e versão para o nível mais granular de resultado a ser resgatado.

No contexto desta implementação, cada fonte de dados tem sua materialização representada por duas tabelas: uma destinada à extração e outra destinada à integração. Nas duas tabelas, há uma família de colunas, identificadores de coluna para representar atributos como volume e tempo de processamento, além de um último com a efetiva carga útil do conteúdo materializado. A chave de cada linha da tabela é calculada por uma função de *hash* sobre o identificador da fonte de dados, a data de extração e o conteúdo. Um ponto interessante neste modelo de dados é a capacidade inerente de criação de rótulos temporais (*timestamp*). Esta característica facilitou a implementação da análise da frequência de atualização do conteúdo da fonte de dados. Outra facilidade utilizada a partir do estudo dos parâmetros de configuração do HBase foi a possibilidade de executar a manutenção do repositório (remoção de conteúdos com idade superior ao determinado pelo administrador do sistema) por meio de um simples atributo de configuração do SGBD chamado TTL (*time to live*).

4.3.3 PENTAHO DATA INTEGRATOR

Diferente do resultado da pesquisa sobre artefatos computacionais capazes de lidar com a questão de virtualização, há vários programas de código aberto disponíveis na comunidade que utilizam uma abordagem de materialização para enfrentar os problemas postos pela integração de dados. Neste trabalho, foi escolhido o Pentaho Data Integrator (PDI) - versão 5.3.0.0-513 Community (PENTAHO, 2016). A escolha deste programa se baseou na simplicidade de sua interface gráfica para modelar processos de integração, na capacidade de estender suas funcionalidades por meio de módulos em Java e Javascript e na familiaridade do autor com tal programa.

A construção do processo de integração no PDI se dá por meio de transformações (*transformations*) e tarefas (*jobs*).As transformações representam o encadeamento das manipulações necessárias em um conteúdo de tal sorte a adequá-lo a um resultado esperado. Já as tarefas representam o encadeamento destas transformações em uma linha do processo de integração, podendo também executar outras tarefas secundárias ou de suporte. A Figura 4.12 mostra um exemplo da interface e como as transformações e as tarefas interagem para manipular e adequar um conteúdo para ser consumido por um

determinado sistema.

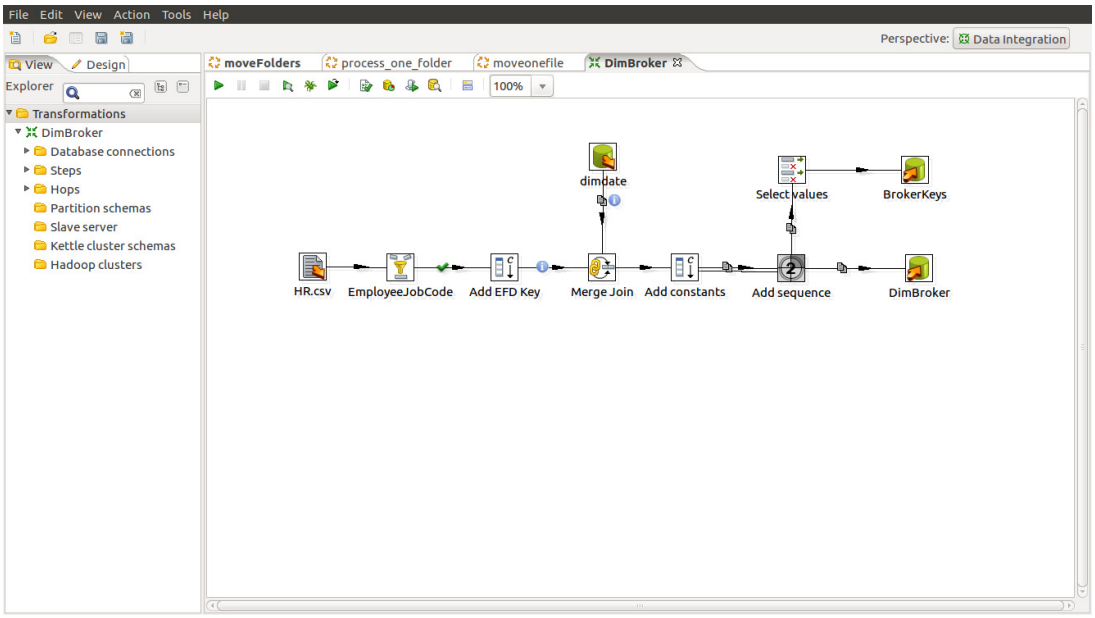

FIG. 4.12: Exemplo Pentaho Data Integrator(PENTAHO, 2016)

Uma vantagem ao se utilizar o PDI é a quantidade expressiva de *plugins* existentes para suportar ações comuns ao processo de integração. Desde tarefas comuns de filtragem e agregação de conteúdos à conexão com diversos tipos de bancos de dados, o programa oferece também a possibilidade de construção de módulos dedicados utilizando as linguagens Java e Javascript. Ele também permite sua execução por meio de linha de comando ou através de outros programas que se utilizem de sua API Java.

Considerando todas estas facilidades, boa parte da implementação dos módulos de materialização e de controle necessários para construir o projeto puderam ser realizadas utilizando apenas o PDI. Somente as funcionalidades dependentes de interação com o administrador do sistema foram resolvidas com outras ferramentas.

4.3.4 COMUNICAÇÃO ENTRE ENTES DO AMBIENTE DE INTEGRAÇÃO

Após tratar da solução de integração, é interessante verificar como a mesma se comunica com as fontes de dados e os sistemas consumidores. Há dois tipos de estrutura de comunicação normalmente empregadas para interligar sistemas: estruturas ponto-a-ponto e estruturas *hub-and-spoke*. Como visto na Seção 2.1.4, a comunicação no primeiro tipo é estabelecida ente a ente, ou seja, cada fonte de dados é ligada ao sistema consumidor que necessita de seu conteúdo. Já no segundo tipo, há um elemento de mediação que conecta as fontes de dados aos sistemas consumidores. A necessidade de um elemento de decisão

que possibilite a adequação da abordagem de integração faz com que a comunicação em *hub-and-spoke* seja a mais indicada no contexto desse trabalho. Contudo, este não é o único motivo para tal escolha.

Naturalmente, a utilização do elemento de mediação leva a um aumento da latência no transporte do conteúdo da fonte de dados ao sistema consumidor. Porém, a falta deste elemento pode transformar o gerenciamento das conexões entre as fontes de dados e os sistemas consumidores em um problema de manutenção ao longo do tempo de vida do ambiente de integração. Como se pode depreender das duas formulações apresentadas no Capítulo 2, na discussão sobre os aspectos do domínio da arquitetura de aplicações, se houver apenas um sistema consumidor, o número de conexões entre as fontes de dados e o sistema consumidor é praticamente o mesmo comparando as estruturas ponto-a-ponto e *hub-and-spoke*. Neste caso, a escolha da forma de comunicação entres os entes só dependerá dos requisitos de latência de recuperação de um determinado conteúdo.

No entanto, ao propor uma solução de integração para um ambiente *Big Data*, presume-se a existência de mais de um sistema consumidor. A medida que a solução de integração sirva a mais e mais sistemas consumidores, o número de conexões em um arquitetura ponto-a-ponto aumenta drasticamente em relação ao esquema de *hub-and-spoke*, tornando-se um desafio para a operação e manutenção deste tipo de ambiente. Logo, a escolha deste tipo de comunicação entre os entes do ambiente de integração não só se alinha à necessidade de implementar uma solução de adequação dinâmica da abordagem de integração, mas também à necessidade de gerenciar o mínimo de conexões entre as fontes de dados e os sistemas consumidores.

A Figura 4.13 mostra o desenho da arquitetura implementada segundo o discutido até o momento. Em um servidor central ficam as funcionalidades de virtualização (Teiid) e controle (PDI). Este servidor é o elemento de mediação desta construção, comunicando-se com as fontes de dados e com os sistemas consumidores, além de controlar todos os aspectos para a seleção da abordagem de integração mais apropriada para cada fonte de dados participante do ambiente. Junto com o servidor central está um servidor dedicado ao repositórios de extração e integração da solução, suportados pelo sistema gerenciador de banco de dados HBase. A separação dos repositórios do resto do sistema deve-se ao critério de escalabilidade necessário ao se considerar um ambiente *Big Data*, isolando possíveis problemas de desempenho e espaço do servidor central.

FIG. 4.13: Implementação da FlexDI

5 TESTES E RESULTADOS

Uma vez determinados o método para seleção de abordagens de integração e a arquitetura capaz de suportá-la dinamicamente, faz-se necessário a realização de testes de tal sorte a confirmar ou refutar as hipóteses levantadas no início deste trabalho. Este capítulo descreve o conjunto de dados idealizado e as dificuldades de sua geração, a construção do ambiente de testes, a criação dos cenários de teste e, finalmente, os resultados obtidos para cada cenário executado.

5.1 TESTES

Idealmente, os testes deveriam simular o processo de integração contínua de um conjunto de dados que descrevessem uma situação do mundo real e que, ao mesmo tempo, apresentassem também as características usualmente encontradas em ambientes *Big Data*: volume, velocidade e variedade. Contudo, a criação do ambiente de simulação idealizado provou-se desafiador. As subseções seguintes descrevem os passos executados, os problemas e as soluções encontradas no sentido de se aproximar ao ambiente de testes desejado.

5.1.1 CONJUNTO DE DADOS

O primeiro ponto trabalhado para criar a simulação foi obter um conjunto de dados que modelasse um aspecto do mundo real, o que geralmente implica a necessidade de algum tipo de esforço de integração, alvo deste trabalho. Inicialmente foi analisada a possibilidade de utilização de dados reais de uma determinada entidade que possuísse um ambiente de integração com as características desejadas de volume, variedade e velocidade. Porém, as restrições ao acesso tornaram-se um impeditivo para seu uso. Sendo assim, a pesquisa focou nos geradores de dados fictícios disponíveis.

Mutos geradores de dados foram encontrados durante a pesquisa, mas a maioria deles apresentava um problema inerente: não reproduziam o modelo de uma situação do mundo real. Ou seja, eram meros geradores de bytes, sem um modelo ou compromisso com seu significado real. Logo não seriam um desafio para a simulação de um processo de integração. Mesmo aqueles produzidos com um esquema para avaliar ambientes *Big Data*

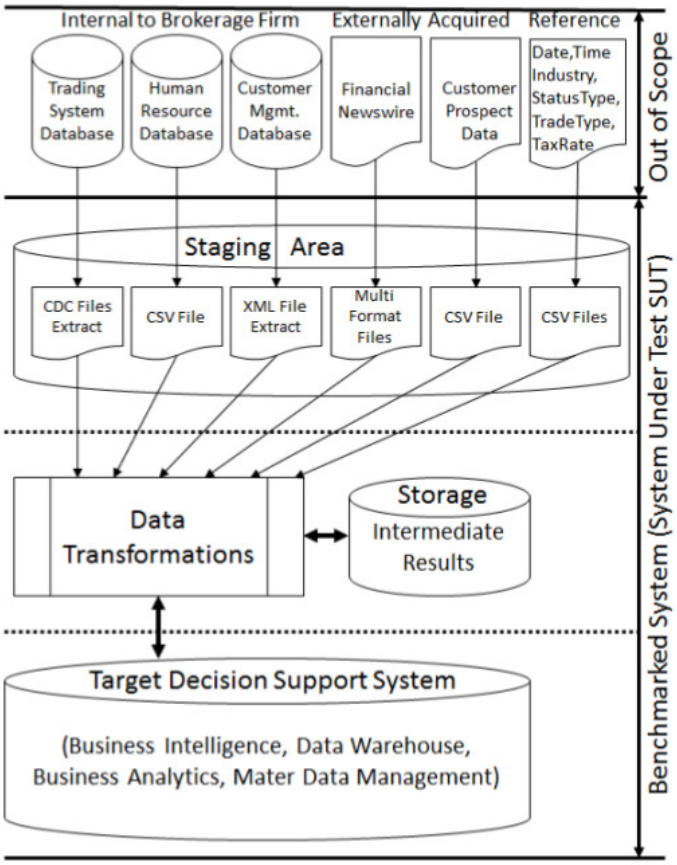

FIG. 5.1: Fluxo do Processo de Comparação(POESS et al., 2014)

(GHAZAL et al., 2013)(COOPER et al., 2010)(HUANG et al., 2011) não se adequavam ao experimento pretendido. Neles, não havia a necessidade de manipulação de fontes de dados de tal sorte a produzir um novo sistema. Os conjuntos de dados fornecidos por tais geradores já representavam o final de um provável processo de integração. Porém, neste esforço de pesquisa, foram encontrados os geradores de dados fornecidos pelo *Transaction Processing Performance Council (TPC)*. Dentre os padrões de comparação disponíveis, um deles mostrou-se alinhado com as expectativas de criação do ambiente de simulação: o TPC-DI (Transaction Processing Performance Council - Data Integration) (POESS et al., 2014).

Segundo os desenvolvedores do processo de comparação de ferramentas de integração, o TPC-DI modela a operação de uma corretora de valores, combinando e transformando os dados de um sistema de processamento de transações em tempo real (*Online Transaction Processing - OLTP*) para criar um *datawarehouse*. A Figura 5.1 mostra o conceito geral do *benchmark* .

A parte superior da Figura 5.1 representa os sistemas que proveem os conteúdos necessários para criação do sistema de suporte a decisão da corretora de valores imaginada.

Porém, ao invés de criar tais sistemas, o TPC disponibiliza um programa de geração (*DIGen*) que produz as extrações necessárias, sendo possível, a partir dele, escalar o volume de cada uma delas. São geradas dezoito diferentes fontes de dados, cada uma provendo uma determinada informação para popular o *datawarehouse*. E estas extrações estão divididas por função: algumas fontes de dados são responsáveis pela população de dados históricos, enquanto outras são dedicadas à atualização dos dados no sistema alvo. Todas estas fontes de dados são geradas para um repositório temporário (*staging area*), algo comum em várias soluções de integração. Após a geração, estas fontes são manipuladas de tal sorte a se adequar ao esquema do *datawarehouse*. O processo termina quando todas as atividades de integração são finalizadas no sistema de suporte a decisão.

O processo de comparação do TPC-DI é dividido em 5 fases: preparação, geração, carga histórica, carga incremental e de auditoria. As fases de preparação e geração são dedicadas à preparação das fontes de dados. Nestas fases, são criadas as fontes de dados responsáveis para a carga histórica e para as cargas incrementais, que, por padrão, são dois volumes representando a atualização de dois dias dos dados históricos. Esta fase não faz parte do processo de comparação e, portanto, não é cronometrada. Na próxima fase, as fontes de dados criadas para a carga histórica são manipuladas de tal maneira a se adequar ao esquema do *datawarehouse* e cumprir as regras de negócio estipuladas. O mesmo processo é aplicado nas cargas incrementais. Este conjunto de fases representa o núcleo do processo de comparação, onde o número total de linhas integradas e o tempo total para adequar os conteúdos das fontes de dados ao esquema do *datawarehouse* são medidos. Segundo preconizado pelo TPC-DI, a métrica de comparação é o quociente do número total de linhas pelo tempo total necessário para realizar toda a integração. Na última fase, são avaliadas a correção e a coerência do resultado que se encontra no sistema de suporte a decisão.

5.1.2 MODELAGEM DA SIMULAÇÃO

De fato, os processos e procedimentos definidos pelo TPC-DI alinham-se aos objetivos buscados para a simulação proposta. Porém, com o estudo e a análise tanto dos procedimentos quanto do gerador de dados fornecido mostraram duas grandes limitações: a falta de variedade nos formatos e repositórios das fontes de dados e o número limitado de cargas incrementais ao longo de todo o processo de integração. A primeira limitação pode ser contornada utilizando um programa como o próprio *Pentaho Data Integrator (PDI)* para transformar a fonte de dados original em uma versão com modelo lógico, formato

e tipos de repositório diferentes. Contudo, o mesmo não pode ser feito com a limitação de cargas incrementais, uma vez que a criação de cada carga incremental é dependente do esquema e dos parâmetros de configuração colocados pelo próprio TPC-DI, algo que não é aberto para o público em geral. Felizmente, após entrar em contato com os desenvolvedores que criaram o programa de geração, foi possível expandir o conjunto de carga incrementais, sendo que o limitante a partir desta liberação foi o processamento e capacidade de armazenamento do *hardware* responsável por tal tarefa.

Neste ponto, o projeto da simulação de um processo de integração já havia conseguido um dos pontos idealizados: um conjunto de dados que modelasse um cenário do mundo real com a possibilidade de variar o volume de tal sorte a se aproximar daqueles encontrados em ambientes *Big Data*. Com a criação de um artefato que possa transformar o formato, o modelo lógico ou o invólucro do conteúdo da fonte de dados, é possível emular o quesito de variedade presente em ambientes *Big Data*. Porém, o processo colocado pelo TPC-DI não foi capaz de entregar um ponto: a velocidade. Esta velocidade deve ser entendida no contexto da integração de dados como a velocidade tanto da disponibilização de novos conteúdos pelas fontes de dados quanto da ingestão pelos sistemas consumidores. No processo descrito pelo TPC-DI, as fonte de dados são geradas de uma única vez, não havendo a preocupação de se ter uma frequência de verificação para saber se uma determinada fonte de dados está ou não disponível. Da mesma forma, no processo do TPC-DI, o sistema consumidor não verifica a disponibilidade para consumo, é a própria solução de integração que é a responsável pela inserção dos dados no sistema de apoio a decisão.

Desta forma, a simulação montada estendeu o processo definido pelo TPC-DI. Ela realiza a entrega periódica de conteúdos das fontes de dados ao longo do processo de integração, assim como um consumo frequente e cadenciado dos sistemas consumidores. Para tanto, foram desenvolvidos dois artefatos computacionais para emular tal situação: um para simular a disponibilização dos conteúdos das fontes de dados e outro para simular o consumo dos conteúdos integrados. Os dois artefatos foram produzidos com a utilização do próprio PDI. No primeiro, o processo criado com o PDI lê os diretórios criados pelo TPC-DI, transforma-os, se necessário, e os coloca nos repositórios escolhidos com uma determinada periodicidade, emulando assim o intervalo de atualização do conteúdo da fonte de dados (t_a^{FD}), como visto nos capítulos anteriores. Do outro lado, foi construído o processo de carga dos dados no *datawarehouse* conforme especificado no TPC-DI. Havia duas formas de implementá-lo: ou a solução de integração provia apenas os dados brutos para serem ainda transformados ou o consumo seria direto, apenas como um processo

de transferência entre a solução de integração e o sistema consumidor. Ou seja, neste último caso, as manipulações necessárias para a carga do *datawarehouse* especificado pelo padrão seriam realizadas pela solução de integração proposta. Dentre as duas opções, optou-se pela segunda, exatamente por ser o pior caso para avaliação das capacidades e das limitações da solução de integração implementada.

5.1.3 AMBIENTE DE TESTE

Com um conjunto de dados escolhido para simular uma situação do mundo real, o próximo passo foi identificar e analisar os recursos necessários para construir o ambiente para testá-lo, sempre tendo em mente quais são os objetivos dos testes: a medição de tráfego entre os entes do ambiente de integração e a necessidade de intervenção de um administrador. Este último objetivo é avaliado a partir do número de ocorrências de mudança na abordagem de integração e na frequência de verificação de novos conteúdos por parte da solução de integração (f_V^{SI}).

A medição de tráfego de dados nos enlaces de comunicação entre os entes do ambiente de integração indica a necessidade da configuração de cada um deles em servidores diferentes, pois não há outra alternativa para realizá-la. Para implementar a solução de integração proposta no capítulo anterior, são necessários dois servidores: um central, que gerencia o processo de integração e a escolha da abordagem de integração adequada para cada fonte de dados, e um auxiliar destinado a ser o repositório de extração e integração dos conteúdos materializados. Além disso, considerando o processo de teste sugerido pelo TPC-DI, existe um sistema de apoio a decisão a ser populado. Logo, há a necessidade de um novo servidor para que seja possível a medição do tráfego na comunicação deste com a solução de integração. Neste mesmo sentido, o modelo do ambiente de integração descrito pelo mesmo processo de *benchmark* pressupõe a extração de dados de seis diferentes sistemas, podendo então residir em seis servidores diferentes. Sendo assim, o ambiente de simulação necessitaria, inicialmente, de nove servidores conectados em rede e a medição do volume de tráfego em oito diferentes enlaces de comunicação com a solução de integração. Estes conjuntos de servidores e enlaces poderiam ser tanto reais como virtuais, pois não foram encontradas restrições neste sentido e o procedimento de teste descrito no TPC-DI considera estas duas opções válidas para construção do ambiente de teste.

Considerando os recursos disponíveis,a velocidade e a disponibilidade para implementação, optou-se pela criação de um ambiente virtual de teste. Foi utilizado um servidor Dell PowerEdge r410, com 2 processadores de seis núcleos, 32GB de RAM e 4TB de

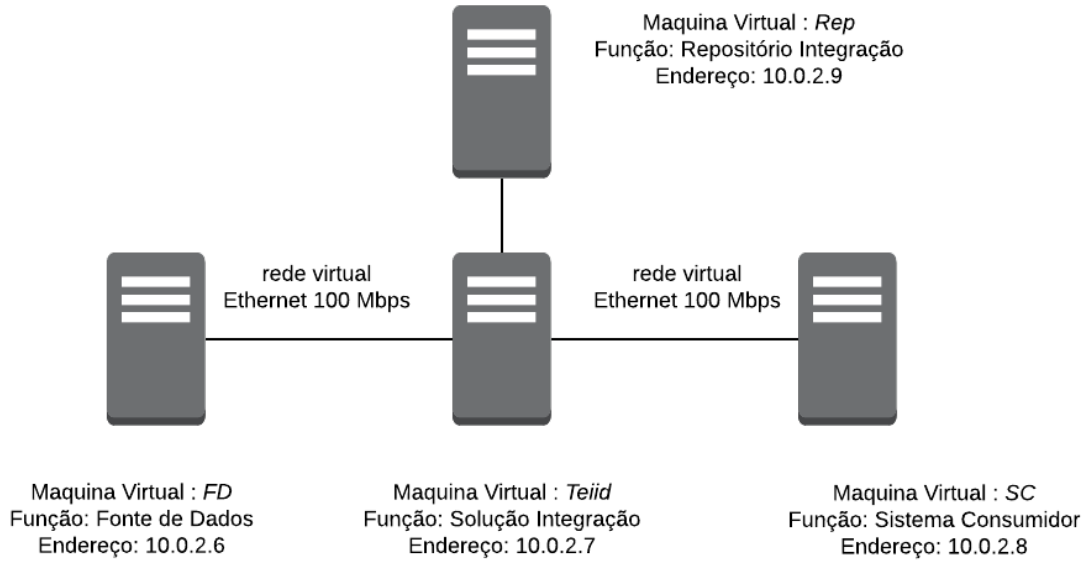

FIG. 5.2: Representação do Ambiente Virtual de Teste

armazenamento, rodando o sistema operacional Linux CentOS 6.7 e o programa Oracle VirtualBox para a criação das máquinas virtuais. Porém, este programa de virtualização possui uma limitação no número de máquinas virtuais a serem criadas: apenas quatro podem ser criadas pela sua interface, podendo alcançar oito máquinas virtuais caso o usuário tenha o conhecimento necessário para realizá-lo por meio de linha de comando. Dado o tempo disponível para explorar as alternativas por meio da linha de comando, optou-se por construir o ambiente de teste com apenas quatro máquinas virtuais. Todas elas estão conectadas por uma rede Ethernet virtual de 100 Mbps. A Figura 5.2 mostra a configuração final do ambiente virtual de teste.

O servidor de virtualização Teiid e os processos de materialização e controle construídos no PDI estão instalados na máquina virtual *Teiid*. Nela, também está o *script* de monitoração do volume sendo transportado entre os entes do ambiente. Como o sistema operacional utilizado é o Linux CentOS, este *script* foi construído com o auxílio do PDI para ativar o programa *tcpdump*, responsável pela coleta do volume de dados transportado entre os servidores. Já o sistema gerenciador de banco de dados HBase está instalado na máquina virtual *Rep*. A máquina virtual *SC* representa o sistema consumidor, onde o processo de consumo criado no PDI está instalado e insere o conteúdo integrado diretamente no esquema do *datawarehouse* criado no SGBD PostgreSQL instalado na mesma máquina. Finalmente, as fontes de dados gerados pelo programa disponibilizado pelo

TAB. 5.1: Configuração das Máquinas Virtuais

Maquina Virtual	**Processadores**	**Memória RAM**	**Disco**
Teiid	4 GB	2	300 GB
FD	16 GB	4	1200 GB
SC	4 GB	2	1000 GB
Rep	4 GB	2	1000 GB

TPC-DI repousam na máquina virtual *FD*. Junto com as fontes de dados, está o processo criado no PDI para simular o surgimento de um novo conteúdo em um repositório pré-determinado. As configurações de cada máquina virtual são mostradas na tabela 5.1. É possível notar que boa parte da memória disponível está alocada na máquina virtual *FD*. Isto se deve ao fato do gerador disponibilizado pelo TPC exigir bastante de processamento e memória no momento em que se escala o conjunto de fontes de dados em volume e em versões de atualização incremental.

5.1.4 CENÁRIOS DE TESTE

Os cenários de teste foram construídos em torno de três considerações principais: o objeto de comparação, seleção das características das fontes de dados e do ambiente de integração a serem estudadas e o tempo de execução esperado.

Para que seja possível determinar que a seleção de abordagem de integração por meio das características das fontes de dados reduz o volume de tráfego e a intervenção humana no processo de integração, é necessário compará-la com estratégias comuns encontradas na área de integração de dados. Uma delas é a materialização de todos conteúdos, sem alteração frequente dos parâmetros passíveis de otimização como a frequência de verificação de novos conteúdos da solução de integração (f_V^{SI}). Essa estratégia estressa o transporte de conteúdos entre os entes do ambiente de integração, uma vez que todos os conteúdos são extraídos para um repositório temporário, manipulados em seguida e guardados em um repositório de integração, que depois são finalmente ingeridos pelos sistemas consumidores. Devido a este comportamento, a estratégia de materialização total dos conteúdos foi escolhida como linha de base para a comparação.

Outro ponto considerado na construção dos cenários foi a análise das características estáticas e dinâmicas das fontes de dados e do ambiente de integração. Apesar de todos eles serem capazes de alterar o tipo de abordagem de integração ou de otimizar os parâmetros verificação de novos conteúdos, é necessário avaliar se cada um deles é passível de verificação quando comparados com a abordagem de integração fixa em materialização.

Como colocado no Capítulo 3, a ocorrência de mudança dos aspectos estáticos ao longo do tempo de vida do ambiente de integração é considerada baixa. Além disso, caso algumas destas características (comportamento passivo da fonte de dados, capacidade de resposta da fonte de dados, existência de sintaxe e de modelo lógico do conteúdo) seja alterada de tal sorte a não possibilitar a virtualização, a comparação com uma abordagem fixa em materialização fica prejudicada. Dessa forma, estes aspectos estáticos não foram considerados na construção dos cenários de testes. Da mesma maneira, a alteração do tempo de vida do conteúdo(t_v^{SI}), do intervalo de atualização do conteúdo(t_a^{SI}) e da probabilidade de perda admitida ($p(x = 0)^{SI}$) na solução de integração alterariam apenas a frequência de verificação de novos conteúdos da solução de integração, algo ligado apenas à otimização do processo de materialização do conteúdo. O ajuste destes parâmetros pode indicar uma redução na intervenção humana, porém não alteraria o volume de dados trafegado na rede que interliga os entes do ambiente de integração de tal forma a ser objeto de comparação com a linha de base selecionada.

Considerando estes argumentos em torno das características da fonte de dados e do ambiente de integração, restam seis que podem ser utilizadas para gerar cenários de teste relevantes para a investigação da alternância de abordagens de integração e da necessidade de intervenção humana no processo: o volume(v_{CNT}), o tempo de vida(t_v^{FD}) e o intervalo de atualização(t_a^{FD}) do conteúdo na fonte de dados, a frequência de verificação(f_V^{SC}) e a probabilidade de perda admitida ($p(x = 0)^{SC}$) pelo sistema consumidor e, por último, a latência da rede de comunicação (t_L).

O último ponto analisado para a criação dos cenários foi o tempo adequado para executar cada um deles. Como cada cenário simula um processo de integração contínua, o tempo total para cada cenário é função do tempo necessário para a carga histórica e para as cargas incrementais fornecidas pelo TPC-DI. Enquanto que o tempo para carga histórica é determinado somente pelo volume gerado para tal ação, o tempo necessário para as cargas incrementais depende não só do volume gerado, mas também do número de rodadas de integração a serem realizadas. Além disso, a simulação do intervalo de atualização da fonte de dados (t_a^{FD}) e da frequência de verificação de novos conteúdos pelo sistema consumidor (f_V^{SC}) também são determinantes, pois a cadência de cada um determina o quão lento ou rápido cada rodada de integração será executada. Dessa forma, o tempo total de teste para cada cenário pode ser determinado algebricamente por:

$$t_{TOTAL} = t_{CH} + [max(t_a^{FD}, f_V^{SC}) + t_{CI}] * N \tag{5.1}$$

onde:

$$N = \text{número de rodadas de integração},$$
$$t_a^{FD} = \text{intervalo de atualização do conteúdo na fonte de dados},$$
$$f_V^{SC} = \text{frequência de verificação de novos conteúdos pelo sistema consumidor},$$
$$t_{CH} = \text{tempo necessário para carga do conteúdo histórico},$$
$$t_{CI} = \text{tempo necessário para carga do conteúdo incremental},$$
$$t_{TOTAL} = \text{tempo total esperado para execução de cada cenário}$$

Uma vez que o ambiente de testes não foi concebido como uma plataforma em produção, a supervisão humana precisa ser permanente. Sendo assim, o tempo de execução de um cenário foi pensado para que não excedesse um tempo total de doze horas. Alguns resultados empíricos mostraram que um intervalo de quinze minutos como máximo entre o intervalo de atualização da fonte de dados e a cadência de verificação de novos conteúdos pelo sistema consumidor e um número máximo de cargas incrementais igual a trinta seriam adequados para a construção do cenários.

Mesmo com a limitação imposta pelas considerações feitas anteriormente, ainda existiam muitos cenários possíveis para serem executados. Devido a limitações de duração deste trabalho, somente três conjuntos de teste foram possíveis de serem realizados. Considera-se que seus aspectos estáticos estão configurados de tal forma a possibilitarem a virtualização dos conteúdos, como mostra a tabela 5.2.

TAB. 5.2: Aspectos Estáticos - Cenários de Teste 1, 2 e 3

Atributo	Descrição	Valor
cPass	Comportamento Passivo do Invólucro	True
cResp	Capacidade de Resposta do Invólucro	True
eStxe	Existência de Sintaxe no Conteúdo	True
eMlog	Existência Modelo de Lógico no Conteúdo	True

No primeiro cenário, foi escolhida a variação do volume do conteúdo da fonte de dados para comparar o comportamento do tráfego de dados entre os entes do ambiente de interação e os ajustes automáticos realizados pela solução sde integração. Para este cenário, foram criados cinco sub-cenários para variação do volume total dos conteúdos das fontes de dados, iniciando com 50 GB e terminando com 1TB. As tabelas 5.3 e 5.4 mostram a configuração deste conjunto de testes.

TAB. 5.3: Aspectos Dinâmicos - Cenário 1 (C1)

Atributo	Descrição	Entidade	Valor
t_v^{FD}	Tempo de Vida do Conteúdo	Fonte de Dados	3600 s
i_a^{FD}	Intervalo de Atualização do Conteúdo	Fonte de Dados	900 s
i_V^{FD}	Intervalo de Verificação do Conteúdo	Fonte de Dados	60 s
i_V^{SC}	Intervalo de Verificação do Conteúdo	Sistema Consumidor	300 s
t_v^{SI}	Tempo de Vida do Conteúdo	Solução Integração	1800 s
$p(x=0)^{FD}$	Probabilidade de Perda	Fonte de Dados	1%
$p(x=0)^{SC}$	Probabilidade de Perda	Sistema Consumidor	1%

TAB. 5.4: Aspectos Dinâmicos - Sub-Cenários do Cenário 1 (C1)

Sub-Cenários	**C1A**	**C1B**	**C1C**	**C1D**	**C1E**
volume (v_{CNT})	50GB	100 GB	250 GB	500 GB	1000 GB

O segundo cenário lida com a variação do tempo de vida do conteúdo na fonte de dados. Diferente do primeiro cenário, a variação ocorre em tempo de execução do processo de integração. O intuito deste teste é estressar as capacidades de virtualização, diminuindo o tempo de vida do conteúdo até um mínimo de sessenta segundos até a metade do processo de integração. Uma vez passada a metade do processo de integração, seu valor volta a aumentar até chegar ao seu valor inicial na última rodada. O plano de variação é mostrado no gráfico 5.3 e a configuração do cenário é mostrada na tabela 5.5.

O último cenário lida com a variação do intervalo de atualização do conteúdo na fonte de dados, da mesma maneira que o segundo cenário. Ou seja, este terceiro cenário estressa as possibilidades de virtualização do conteúdo de uma fonte de dados em tempo de execução, reduzindo o intervalo de atualização para um mínimo e depois retornando ao seu valor original. O plano de variação e a configuração do cenário são mostrados, respectivamente no gráfico 5.4 e na tabela 5.5.

TAB. 5.5: Aspectos Dinâmicos - Cenários 2 (C2) e 3 (C3)

Atributo	Descrição	Entidade	Valor
t_v^{FD}	Tempo de Vida do Conteúdo	Fonte de Dados	3600 s
i_a^{FD}	Intervalo de Atualização do Conteúdo	Fonte de Dados	900 s
i_V^{FD}	Intervalo de Verificação do Conteúdo	Fonte de Dados	60 s
i_V^{SC}	Intervalo de Verificação do Conteúdo	Sistema Consumidor	300 s
t_v^{SI}	Tempo de Vida do Conteúdo	Solução Integração	1800 s
$p(x=0)^{FD}$	Probabilidade de Perda	Fonte de Dados	1%
$p(x=0)^{SC}$	Probabilidade de Perda	Sistema Consumidor	1%
v_{CNT}	Volume do Conteúdo	Fonte de Dados	250 GB

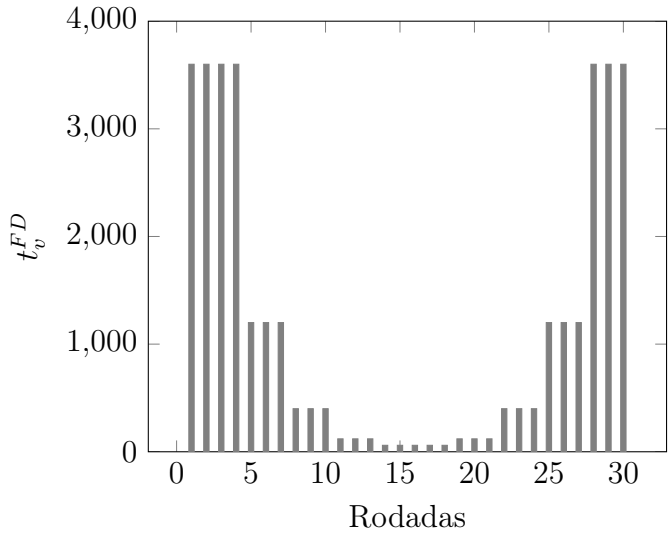

FIG. 5.3: Avaliação da Seleção de Abordagens de Integração - Cenário2 - Plano de Variação do Tempo de Vida do Conteúdo

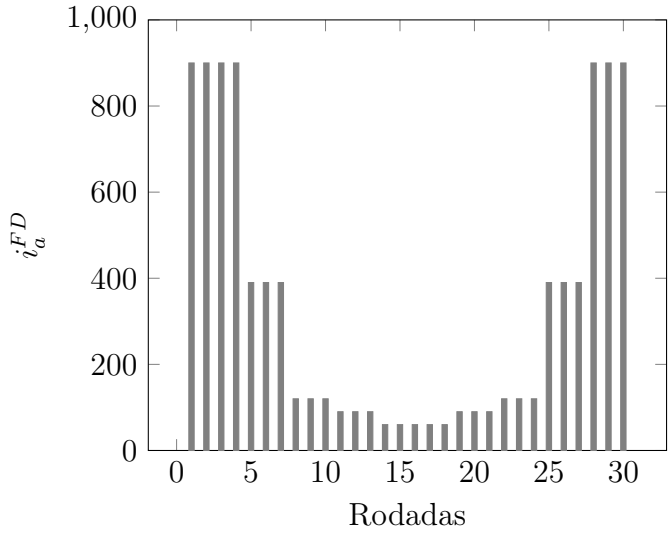

FIG. 5.4: Avaliação da Seleção de Abordagens de Integração - Cenário2 - Plano de Variação do Intervalo de Atualização do Conteúdo

5.2 RESULTADOS E ANÁLISES

Os resultados da execução dos cenários discutidos na seção anterior são mostrados e discutidos a seguir:

5.2.1 CENÁRIO 01

Os gráficos das figuras 5.5 e 5.6, mostram o volume de dados tráfego quando a abordagem de integração é híbrida ($C1$) em comparação à linha de base (LBS) que possui uma abordagem de integração fixa em materialização. Além disso, são mostradas as ocorrências de ajuste pela solução de integração com abordagem de integração híbrida, sejam para a seleção de uma abordagem mais adequada (ABI), sejam para ajuste da frequência de verificação de novos conteúdos (f_V^{SI}).

Analisando os resultados deste cenário, nota-se que a medida que o volume do conteúdo da fonte de dados aumenta, as possibilidades de virtualização ficam reduzidas, chegando ao seu pior desempenho quando o volume total trafegado chegou a 1 TB. A tabela 5.6 mostra as reduções incrementais e totais em relação ao volume trafegado quando da utilização da abordagem de materialização (linha de base).

TAB. 5.6: Comparação de volume trafegado em relação à linha de base - Cenário 1

	C1A	**C1B**	**C1C**	**C1D**	**C1E**
Redução Incremental	31.132%	23.749%	9.308%	6.320%	4.083%
Redução Total	11.129%	8.286%	3.247%	2.205%	1.424%

Porém, apenas debitar a impossibilidade de virtualização ao volume crescente do conteúdo mascara o efeito de outras características do ambiente de integração. Para evitar tal falha de interpretação, faz-se necessário retornar às condições de contorno colocadas nas equações (3.14) e (3.15). A equação (5.2) mostra a expansão da condição em seu ponto de transição, enquanto a equação (5.3) mostra a expansão do tempo de transporte do conteúdo entre a fonte de dados e o sistema consumidor

$$\begin{aligned} t_v^{FD} - t_T^{FDSC} &= t_a^{FD} \\ t_T^{FDSC} &= t_v^{FD} - t_a^{FD} = \text{constante} \end{aligned} \tag{5.2}$$

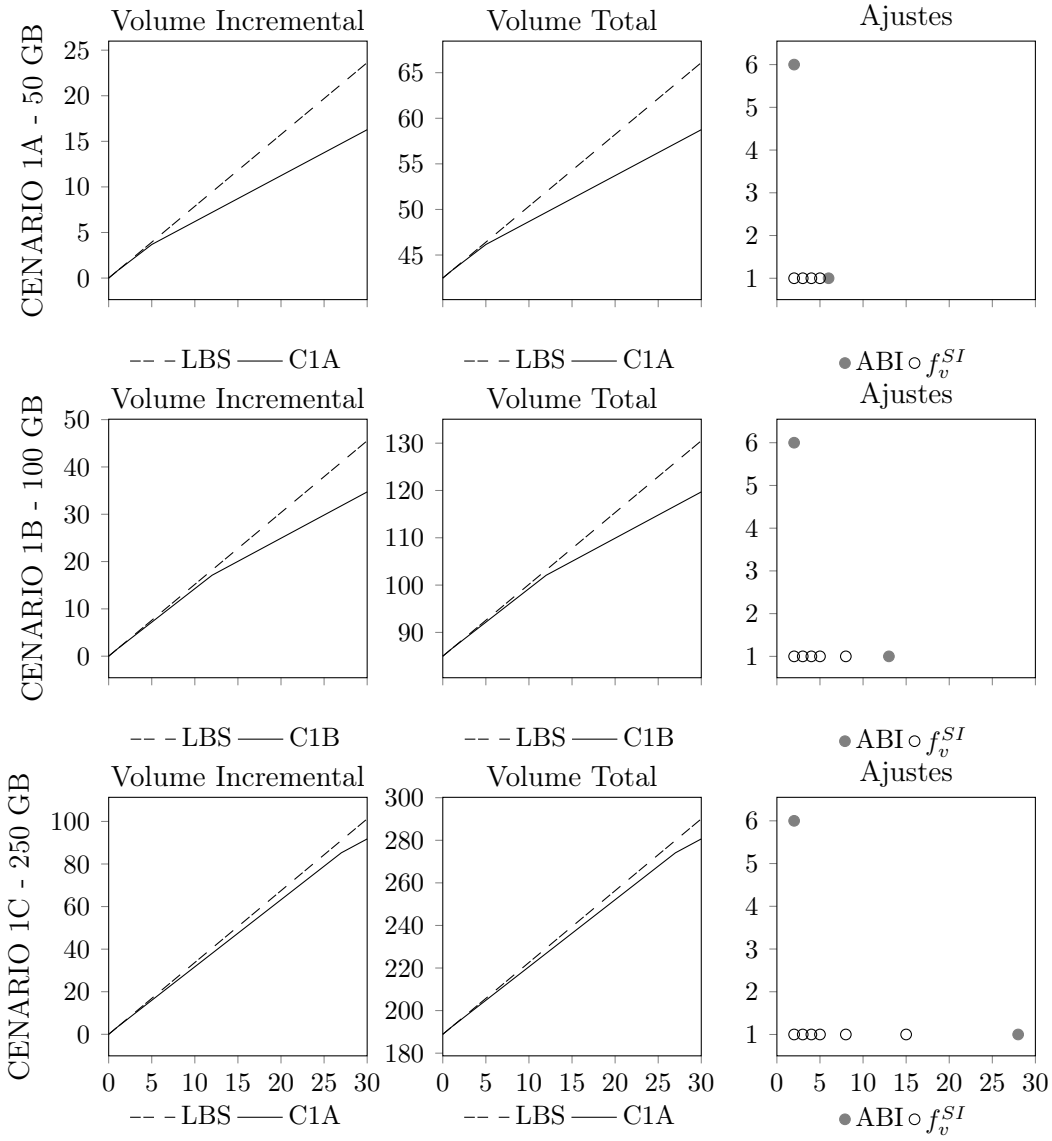

FIG. 5.5: Avaliação da Seleção de Abordagens de Integração - Cenários 1A,1B e 1C

$$t_T^{FDSC} \approx t_T^{FDSI} + t_T^{SISC} + t_p = \\ = v_{CNT} * l_T^{FDSI} + v'_{CNT} * l_T^{SISC} + t_p \tag{5.3}$$

A equação (5.2) mostra que, para este teste, o tempo de transporte deve ser uma constante. Já a equação (5.3) exibe a dependência do tempo de transporte com o volume do conteúdo sendo integrado, a latência das conexões das fontes de dados e dos sistemas consumidores com a solução de integração e com o tempo de processamento do conteúdo.

Nota-se que o único parâmetro que pode ser ajustado para manter o tempo de transporte do conteúdo entre a fonte de dados e o sistema consumidor constante é o tempo de

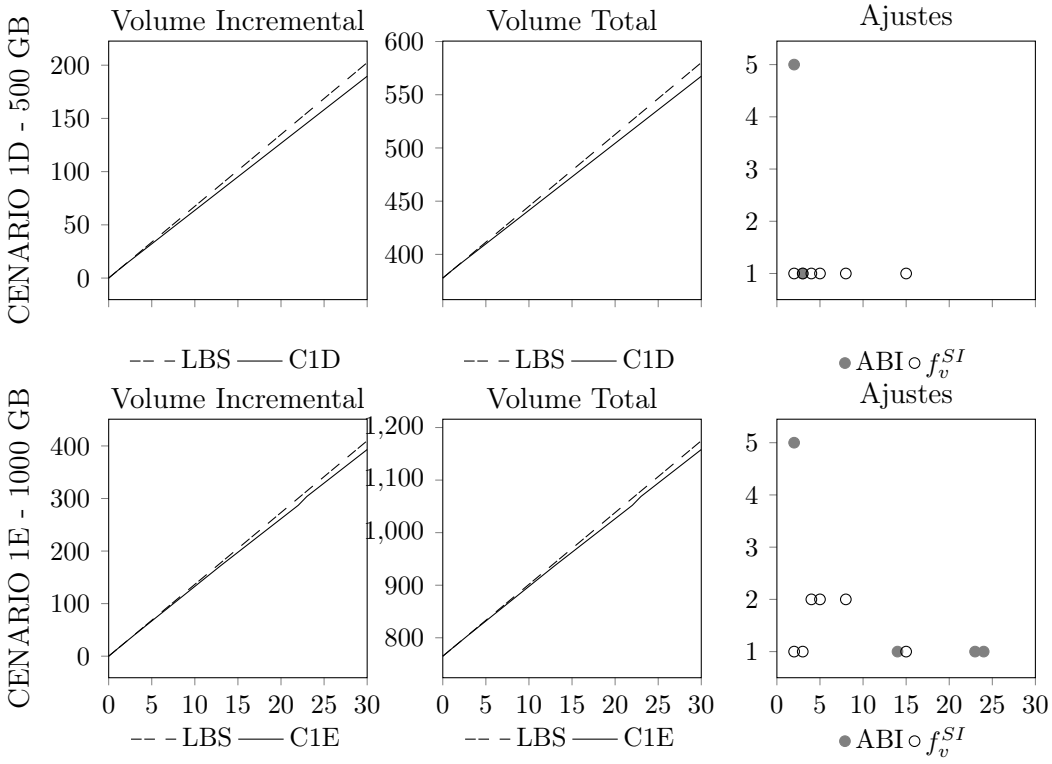

FIG. 5.6: Avaliação da Seleção de Abordagens de Integração - Cenários 1D e 1E

processamento, uma vez que todos os outros estão, a princípio, fora do escopo de administração da solução de integração. Apesar do tempo de processamento necessário para adequar o conteúdo da fontes de dados ao esquema do sistema consumidor ser dependente também do volume, ele também é um reflexo do desempenho de *hardware* e do *software* onde a solução de integração está instalada. Além disso, este tempo depende da qualidade do encadeamento das manipulações a que o conteúdo estará submetido. Logo, pode-se concluir que não só há possibilidades de virtualização a partir de características que não estão dentro do alcance de otimização da solução de integração, mas também há possibilidades de mantê-la mesmo que estas não permitam, desde que o tempo de processamento possa compensá-las. É importante destacar que a melhoria do tempo para adequar o conteúdo é limitada, não sendo solução para todos os casos onde a virtualização deixa de ser um opção de abordagem de integração.

Outro resultado interessante a ser mostrado é que, nos casos onde a virtualização não foi possível, houve uma tentativa da solução de integração de ajustar a frequência de verificação de novos conteúdos. Esses ajustes automáticos por parte da solução de integração representam a necessidade de intervenção humana no processo de integração considerado como linha de base. Intui-se que estes ajustes permitam um menor consumo de processamento e de memória do servidor onde a solução de integração reside, porém

esta medição está fora do escopo dos cenários de teste.

5.2.2 CENÁRIO 02

O resultado do segundo cenário é mostrado nos gráficos da Figura 5.7.

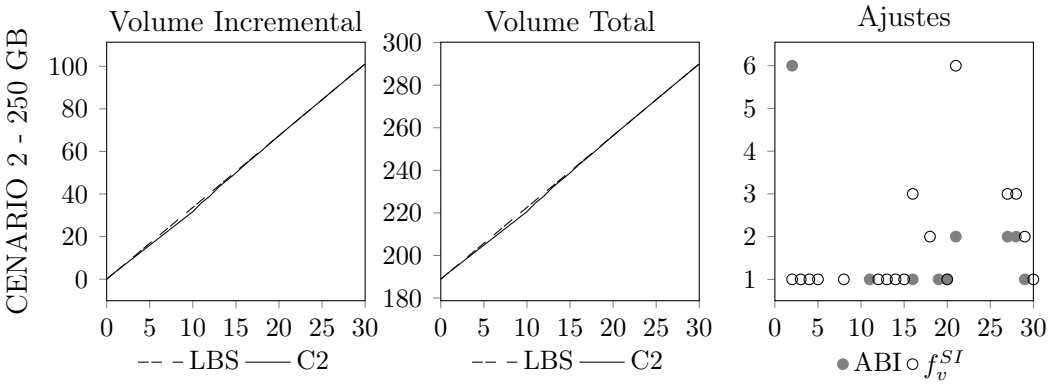

FIG. 5.7: Avaliação da Seleção de Abordagens de Integração - Cenário 2

Enquanto $t_v^{FD} - t_T^{FDSC}$ for maior que $t_a^F D$, a frequência de verificação do sistema consumidor precisa apenas ser igual ou superior a frequência de atualização da fonte de dados. Neste caso, não há condição de perda do conteúdo devido ao seu desaparecimento em seu repositório original, segundo a hipótese do modelo proposto no Capítulo 3.

Porém, a medida que o tempo de vida diminui de tal sorte que $t_a^F D > t_v^{FD} - t_T^{FDSC}$, o sistema consumidor precisa investigar com mais frequência para que não haja acúmulo de conteúdos a serem integrados, impedindo assim a possibilidade de virtualização. Porém este aumento na cadência de verificação tem um limite. Por exemplo, os agendadores de tarefas de sistemas operacionais como o Windows ou Linux possuem uma frequência máxima de uma verificação por minuto. Dessa forma, se a formulação colocada na equação (3.15) estipular uma frequência maior, será necessário que o conteúdo seja materializado para posterior consumo.

O resultado apresentado na Figura 5.7 mostra este comportamento. Considerando que o tempo de vida do conteúdo da fonte de dados é reduzido ao longo do processo de integração, seu valor aproxima-se do seu intervalo de atualização (900 s). Neste sentido, a frequência de verificação de novos conteúdos pelo sistema consumidor é afetado pela segunda condição mostrada na equação (3.14), não sendo mais suficiente ser apenas superior à frequência de atualização do conteúdo, chegando a um limite (neste caso, uma verificação a cada 300 segundos) que não pode ser alterado senão pelo responsável do sistema consumidor.

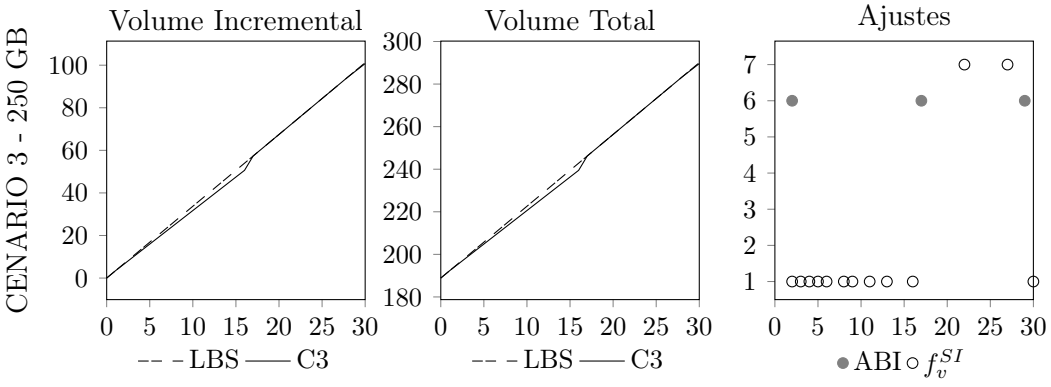

FIG. 5.8: Avaliação da Seleção de Abordagens de Integração - Cenário 3

Este cenário poderia ser amenizado de duas formas: uma, por meio da otimização do tempo de processamento(t_p) da mesma forma que sugerido no cenário anterior. Outra forma de relaxar as condições para permitir a virtualização de um conteúdo é ajustar a probabilidade de perda admitida pelo sistema consumidor ($p(x = 0)^{SC}$), que pode ser diferente para cada uma das fontes de dados. Há fontes que podem admitir perdas maiores do que outras. Contudo, este ajuste pode ser limitado, pois quem define esta perda é o consumidor do conteúdo. Ou seja, pode estar fora do escopo de otimização a ser efetuado pelo administrador da solução de integração.

5.2.3 CENÁRIO 03

O resultado do último cenário é mostrado na Figura 5.8, que mostra a variação do intervalo de atualização da fonte de dados, considerando o mesmo volume de 250 GB do cenário anterior. Neste caso, ao diminuir o intervalo de atualizações de tal forma a ser inferior ao tempo de vida da fonte de dados subtraindo o tempo de transporte, a probabilidade de integrar o conteúdo é de 100%. Aparentemente, reduzir o intervalo de atualização do conteúdo da fonte de dados aumentaria as possibilidades de virtualização, uma vez que, pela primeira condição da equação (3.14), bastaria que a frequência de verificação do sistema consumidor fosse superior à frequência de atualização dos conteúdos das fontes de dados.

Porém, esta ação diminui, por ela mesma, as condições de virtualização, bastando novamente resgatar a equação (3.15). Ao diminuir o intervalo de atualização, aumenta-se a frequência de atualização ($f_a^{FD} = 1/t_a^{FD}$). Aumentar a frequência de atualização do conteúdo significa, em determinadas condições, aumentar a frequência de verificação do sistema consumidor. E, como colocado no cenário anterior, esta frequência tem um limite e está fora do alcance de alteração da administração da solução de integração. O resultado

mostra esta condição, uma vez que o intervalo de verificação pelo sistema consumidor foi configurado em 300 segundos e o plano de variação do intervalo de atualização do conteúdo possui um mínimo de 60 segundos.

Outro detalhe importante a ser destacado do resultado mostrado na Figura 5.8 é a tentativa da solução de integração de otimizar sua frequência de verificação de novos conteúdos, uma vez que a virtualização não é possível. Este comportamento não é só visto neste cenário, mas também em todos aqueles que tiveram suas oportunidades de virtualização frustradas.

5.2.4 SUMÁRIO DAS ANÁLISES

Os resultados mostram a viabilidade de utilizar as características das fontes de dados e do ambiente de integração para selecionar adequadamente as abordagens de integração, além de proporcionar ajustes automáticos da frequência de verificação de novos conteúdos pela solução de integração.

Apesar do baixo desempenho relacionado ao tráfego de dados entre os entes de integração, mostrado na tabela 5.6, o resultado não é só reflexo das características configuradas para as fontes de dados e demais entes do ambiente de integração. A quantidade de rodadas também afetou o resultado final. As trinta rodadas representam a integração de apenas trinta dias de consumo do sistema de apoio à decisão modelado pelo TPC-DI. Espera-se que, para este tipos de sistemas, a integração perdure por mais tempo e, assim, os ganhos relacionados à diminuição do tráfego de dados sejam maiores.

De qualquer forma, há limitações no emprego do método de seleção. A primeira limitação está relacionada ao volume do conteúdo. A medida que ele cresce, o tempo de transporte entre a fonte de dados e o sistema consumidor aumenta e desequilibra a relação entre o tempo de vida e o intervalo de atualização do conteúdo da fonte de dados. Esse desequilíbrio faz com que a frequência de verificação de novos conteúdos pelo sistema consumidor seja incrementada até o ponto que chega a seu limite. Neste limiar, a fonte de dados deixa de ser passível de virtualização.

Outra limitação é percebida na análise dos dois últimos cenários. Nota-se que quando o intervalo de atualização e o tempo de vida da fonte de dados possuem valores próximos, as possibilidades de virtualização diminuem. Isso ocorre, pois esta condição também implica na necessidade de verificações mais constantes pelo sistema consumidor, até chegar a um ponto que não são mais possíveis.

É possível remediar estas situações de duas formas: melhorando a qualidade da rede

de comunicação e o desempenho do processamento dos conteúdos pela solução de integração. A primeira forma depende da capacidade do administrador da solução de integração otimizar ou influenciar a melhoria da taxa de transmissão da rede. Caso seja possível, o tempo de transporte pode ser reduzido e compensar eventuais desequilíbrios ocasionados ou pelas alterações no intervalo de atualização ou pelas alterações decorrentes das alterações no tempo de vida do conteúdo.

A outra forma de compensação pode vir da melhoria do processamento do conteúdo para se adequar ao consumo pelo sistema consumidor. Isso pode ser realizado tanto pela ampliação dos recursos de *hardware* e *software* onde reside a própria solução de integração quanto pela otimização do próprio processo de integração, seja pela seleção correta da ordem de manipulação, seja pelo aprimoramento da própria manipulação.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo final apresenta as conclusões deste trabalho e suas contribuições, assim como as limitações encontradas durante seu desenvolvimento e as propostas de futuros trabalhos que podem estender o conhecimento gerado neste esforço de pesquisa.

6.1 CONCLUSÕES

O objetivo do trabalho foi desenvolver uma sistemática de escolha e adaptação das abordagens de integração apoiada pela caracterização das fontes de dados. O resultado inicialmente esperado ao utilizar tal sistemática era a redução do tráfego de dados entres os entes do ambiente de integração assim como a minimização da intervenção humana no processo de integração.

No Capítulo 3 deste trabalho, foram analisadas as características das fontes de dados levantadas na revisão bibliográfica e sua relevância na escolha da abordagem de integração. Nesta análise, foi identificado que não só os aspectos estáticos destas fontes de dados poderiam indicar a possibilidade de sua virtualização, mas também os aspectos dinâmicos, pouco explorados na literatura, poderiam apoiar tal seleção. Além disso, foi possível associar tais aspectos dinâmicos a uma formulação matemática que permitiu a construção de um fluxo de decisão que, em tempo de execução, seleciona a abordagem mais adequada para uma determinada fonte de dados em um dado instante da existência do ambiente de integração.

Uma vez que o método de decisão foi criado, uma arquitetura híbrida para a solução de integração foi desenvolvida e descrita no Capítulo 4. Neste capítulo, foram discutidos os requisitos necessários para que fosse possível a aplicação da sistemática da escolha de abordagem de integração em tempo de execução de um processo de integração. O resultado da discussão e da análise foi a criação de uma arquitetura para a solução de integração: a FlexDI.

De posse de uma arquitetura capaz de apoiar a seleção dinâmica das abordagens de integração a partir das características das fontes de dados e do ambiente de integração em que estão inseridas, foram realizados testes utilizando os dados e os procedimentos descritos pelo *Transaction Processing Council (TPC)*. Os testes realizados mostraram

que, ao se variar o volume, o tempo de vida e o intervalo de atualização do conteúdo de uma fonte de dados, há uma redução no tráfego de dados entre os entes do ambiente de integração ao se comparar com uma construção tradicional onde os conteúdos de todas as fontes de dados são materializadas. Além disso, a implementação da arquitetura mostrou não só a possibilidade de ajuste automático da abordagem de integração, mas também de outros parâmetros, como a frequência de verificação de novos conteúdos nos momentos onde uma abordagem de materialização estava sendo utilizada. Contudo, há limitações que precisam ser observadas e que, em certas condições, podem ser mitigadas pela melhora da qualidade da rede de comunicação que conecta os entes do ambiente de integração e pelo aprimoramento do processo de integração necessário para adequar os conteúdos ao consumo dos sistemas consumidores.

Conclui-se, a partir dos resultados obtidos, que a seleção de abordagens de integração por meio das caraterísticas das fontes de dados e do ambiente de integração pode reduzir não só o tráfego de dados no ambiente de integração como também a necessidade de intervenção humana no processo de integração.

6.2 CONTRIBUIÇÕES

Este trabalho contribuiu para a pesquisa na área de integração de dados em alguns pontos, levantando novas interpretações e apresentando algumas propostas como :

Levantamento de Aspectos Estáticos e Dinâmicos das Fontes de Dados Neste levantamento foram apontados as características estáticas e dinâmicas capazes de determinar a abordagem de integração para uma fonte de dados. Além disso, foram separadas aquelas que são próprias para serem avaliadas no momento de projeto da solução de integração de dados daquelas que seriam importantes em tempo de execução do processo de integração;

Proposta de um Relacionamento Matemático entre Aspectos Dinâmicos No estudo dos aspectos dinâmicos, foram propostos relacionamentos matemáticos que unem vários conceitos importantes no tempo de execução do processo de integração como o tempo de vida do conteúdo em um determinado repositório e a sua frequência de atualização;

Introdução do Conceito de Probabilidade de Perda de Conteúdo Dentro dos estudos dos aspectos dinâmicos das fontes de dados e de seus inter relacionamentos , pode-se perceber a necessidade da criação de um conceito de probabilidade

de perda de um conteúdo, ou seja, mensurar a possibilidade de um conteúdo não ser integrado dentro de um determinado intervalo especificado;

Proposta de uma Arquitetura Híbrida de Integração com Seleção Dinâmica de Abordagens A partir da proposta de seleção das abordagens de integração a partir de certos aspectos que as fontes de dados possuem, foi criada uma arquitetura híbrida de integração de dados, por meio de mediadores e extratores e utilizando um novo módulo de controle para seleção dinâmica das abordagens, ação pouco explorada na literatura.

6.3 LIMITAÇÕES

Apesar da avaliação sobre a redução do tráfego de dados nas conexões de rede que interligam os elementos do ambiente de integração e da intervenção humana no processo de integração ter sido bem sucedida, algumas limitações foram encontradas durante a execução do trabalho, como se segue:

Falta de um Modelo Apropriado A falta de um conjunto de dados próprio para testar um processo de integração no contexto *Big Data* tornou-se um desafio para a implementação dos testes. A utilização do conjunto de dados provido pelo TPC-DI foi determinante para o sucesso dos testes, porém foram necessários alguns ajustes para que ele pudesse se adequar ao processo de integração idealizado;

Geração do Conjunto de Dados A geração de grandes volumes e de várias cargas incrementais próximas a um terabyte exigiu muito do processamento e da memória do servidores responsáveis, de tal sorte que em alguns momentos esgotaram completamente os recursos de *hardware* e terminaram em um estado não previsto (travamento);

Impacto da Alternância de Abordagens Não foi possível avaliar o impacto da seleção dinâmica das abordagens de integração no consumo de processamento e memória dos servidores da solução de integração dentro do prazo determinado para término deste trabalho devido a falta do conhecimento necessário para resgatar tais informações em máquinas virtuais;

Garantia de Consistência Ainda decorrente do prazo de término do trabalho, não possível implementar a fase de auditoria especificada no processo descrito no TPC-DI para garantir que os conteúdos integrados pela abordagem de integração

fixada em materialização fossem idênticos àqueles integrados pela abordagem de integração flexível.

6.4 TRABALHOS FUTUROS

Durante este trabalho de pesquisa, vários outros pontos de interesse foram identificados, porém não puderam ser tratados adequadamente devido às restrições de escopo e tempo destinados. No inicio da revisão bibliográfica, onde foram identificados os principais eixos de análise das características das fontes de dados (sintático, semântico, estrutural e sistêmico), o tratamento da questão semântica já apresentava um volume de análise que ultrapassava o escopo previsto para este trabalho. Seu tratamento foi resumido por meio da avaliação do tempo de processamento em um ambiente de integração, sendo esta a característica relevante para a seleção de abordagens. Contudo, é necessária uma avaliação mais detalhada sobre o tratamento da heterogeneidade semântica, uma vez que o espectro de tratamento varre um espectro de simples mapeamentos à resolução por meio de ontologias. E cada um pode afetar de forma diferente o tempo de processamento (adequação do conteúdo), permitindo sua virtualização ou não.

Na análise das características estáticas, foi encontrado um projeto da fundação Apache (Apache Tika) para identificação automática dos metadados do conteúdo sendo integrado. Essa ferramenta poderia eliminar a necessidade de indicação da existência de sintaxe e modelo lógico quando do cadastro da fonte de dados na solução de integração por seu administrador. Além disso, seria necessário avaliar se esta automatização influenciaria o tempo total de processamento do conteúdo.

Outro ponto percebido durante a análise das características dinâmicas foi a oportunidade de mudança do modo de processamento (em linha ou paralelo e distribuído) e da forma de ingestão (*batch* ou *streaming*) do conteúdo como reflexo do aumento de seu volume ou da variação de seu intervalo de atualização. Ainda durante esta análise, foi percebida a possibilidade de ajuste ou o aviso ao administrador da solução de integração acerca do espaço destinado aos repositórios de extração e integração por meio da avaliação do tempo de vida e do intervalo de atualização do conteúdo nestes repositórios. Por fim, a forma de seleção da abordagem de integração pode se beneficiar de algoritmos de aprendizado de máquina, melhorando seu desempenho e assertividade no momento da troca.

Finalmente, foram percebidas outras possibilidades de análise durante a execução dos testes. A primeira refere-se à monitoração do consumo de processamento e memória na

solução de integração quando a abordagem de materialização é selecionada. Considerando que a frequência de verificação de novos conteúdos pela solução de integração consome estes recursos, intui-se que, casos as características das fontes de dados e do ambiente de integração permitam a redução desta frequência, o consumo de processamento e de memória devem ser menos exigidos. Já a segunda refere-se à avaliação de resultados dos testes quando se estrangula a banda de rede de comunicação. Foi identificada tal possibilidade na especificação do programa de virtualização (Oracle VirtualBox), porém não houve tempo hábil para sua avaliação dentro do escopo de tempo destinado a este trabalho.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABADI, D. J.; AGRAWAL, R.; AILAMAKI, A.; BALAZINSKA, M.; BERNSTEIN, P. A.; CAREY, M. J.; CHAUDHURI, S.; DEAN, J.; DOAN, A.; FRANKLIN, M. J.; GEHRKE, J.; HAAS, L. M.; HALEVY, A. Y.; HELLERSTEIN, J. M.; IOANNIDIS, Y. E.; JAGADISH, H. V.; KOSSMANN, D.; MADDEN, S.; MEHROTRA, S.; MILO, T.; NAUGHTON, J. F.; RAMAKRISHNAN, R.; MARKL, V.; OLSTON, C.; OOI, B. C.; RÉ, C.; SUCIU, D.; STONEBRAKER, M.; WALTER, T. ; WIDOM, J. The beckman report on database research. **SIGMOD Record**, v. 43, n. 3, p. 61–70, 2014. Disponível em: <http://doi.acm.org/10.1145/2694428.2694441>. Acesso em: Fri, 19 Feb 2016 20:04:00 -0200.

ASHISH, N.; KNOBLOCK, C. A. ; SHAHABI, C. Selectively materializing data in mediators by analyzing source structure, query distribution and maintenance cost. In: PROCEEDINGS OF THE 2ND INTERNATIONAL WORKSHOP ON WEB INFORMATION AND DATA MANAGEMENT, 1., WIDM '99, 1., 1999. **Anais**... New York, NY, USA: ACM, 1999, p. 33–37. Disponível em: <http://doi.acm.org/10.1145/319759.319774>. Acesso em: .

BONDI, A. B. Characteristics of scalability and their impact on performance. In: PROCEEDINGS OF THE 2ND INTERNATIONAL WORKSHOP ON SOFTWARE AND PERFORMANCE, 1., WOSP '00, 1., 2000. **Anais**... New York, NY, USA: ACM, 2000, p. 195–203. Disponível em: <http://doi.acm.org/10.1145/350391.350432>. Acesso em: 2016-04-10.

CAO, Y.; CHEN, Y. ; JIANG, B. A study on self-adaptive heterogeneous data integration systems. In: RESEARCH AND PRACTICAL ISSUES OF ENTERPRISE INFORMATION SYSTEMS II, VOLUME 1, IFIP TC 8 WG 8.9 INTERNATIONAL CONFERENCE ON RESEARCH AND PRACTICAL ISSUES OF ENTERPRISE INFORMATION SYSTEMS (CONFENIS 2007), OCTOBER 14-16, 2007, BEIJING, CHINA, 1., 2007. **Anais**... [S.l.: s.n.], 2007, p. 65–74. Disponível em: <http://dx.doi.org/10.1007/978-0-387-75902-9_7 > $.Acessoem : Fri, 19Feb201620 : 04 : 00 - 0200$.

CHEN, M.; MAO, S. ; LIU, Y. Big data: A survey. **Mobile Networks and Applications**, v. 19, n. 2, p. 171–209, 2014. Disponível em: <http://dx.doi.org/10.1007/s11036-013-0489-0>. Acesso em: Fri, 19 Feb 2016 20:04:00 -0200.

COOPER, B. F.; SILBERSTEIN, A.; TAM, E.; RAMAKRISHNAN, R. ; SEARS, R. Benchmarking cloud serving systems with ycsb. In: PROCEEDINGS OF THE 1ST ACM SYMPOSIUM ON CLOUD COMPUTING, 1., SOCC '10, 1., 2010. **Anais**... New York, NY, USA: ACM, 2010, p. 143–154. Disponível em: <http://doi.acm.org/10.1145/1807128.1807152>. Acesso em: .

DE FARIA CORDEIRO, K. **aDApTA: ADAPTIVE APPROACH FOR INFORMATION INTEGRATION TO SUPPORT DECISION MAKING IN COMPLEX ENVIRONMENTS**. 2015. 129 f. Tese (Doctoral Thesis in Informatics) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

DEROOS, D.; COSS, R. **Hadoop for Dummies**. New Delhi: Wiley Publication, 2014.

DOAN, A.; HALEVY, A. Y. ; IVES, Z. G. **Principles of Data Integration**. [S.l.]: Morgan Kaufmann, 2012. ISBN 978-0-12-416044-6.

DONG, X. L.; SRIVASTAVA, D. Big data integration. **PVLDB**, v. 6, n. 11, p. 1188–1189, 2013. Disponível em: <http://www.vldb.org/pvldb/vol6/p1188-srivastava.pdf>. Acesso em: Fri, 19 Feb 2016 20:04:00 -0200.

THE APACHE SOFTWARE FOUNDATION. Apache Tika. Disponível em: <https://tika.apache.org/>. Acesso em: 23 outubro de 2016.

FOWLER, M. Dealing with roles. In: PATTERN LANGUAGES OF PROGRAMMING (PLOP'97) AND EUROPLOP'97 CONFERENCE, 1., 1997. **Anais**... [S.l.: s.n.], 1997. Disponível em: <https://pdfs.semanticscholar.org/ddb6/0826697eadb18e9864aff0b9d9281733c288.pdf>. Acesso em: .

GHAZAL, A.; RABL, T.; HU, M.; RAAB, F.; POESS, M.; CROLOTTE, A. ; JACOBSEN, H.-A. Bigbench: Towards an industry standard benchmark for big data analytics. In: PROCEEDINGS OF THE 2013 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 1., SIGMOD '13, 1., 2013. **Anais**... New York, NY, USA: ACM, 2013, p. 1197–1208. Disponível em: <http://doi.acm.org/10.1145/2463676.2463712>. Acesso em: .

GIORDANO, A. D. **Data Integration Blueprint and Modeling: Techniques for a Scalable and Sustainable Architecture**. 1st. ed. [S.l.]: IBM Press, 2011. ISBN 0137084935, 9780137084937.

HAREN, V. **TOGAF Version 9.1**. 10th. ed. [S.l.]: Van Haren Publishing, 2011. ISBN 9087536798, 9789087536794.

HOHPE, G.; WOOLF, B. **Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321200683.

HUANG, S.; HUANG, J.; DAI, J.; XIE, T. ; HUANG, B. The hibench benchmark suite: Characterization of the mapreduce-based data analysis. In: AGRAWAL, D.; CANDAN, K. S. ; LI, W.-S. (Org.). **New Frontiers in Information and Software as Services: Service and Application Design Challenges in the Cloud**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 209–228. ISBN 978-3-642-19294-4.

HULL, R.; ZHOU, G. A framework for supporting data integration using the materialized and virtual approaches. **SIGMOD Rec.**, v. 25, n. 2, p. 481–492, 1996. Disponível em: <http://doi.acm.org/10.1145/235968.233365>. Acesso em: .

ISO/IEC/IEEE. Systems and software engineering – architecture description. **ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)**, v. ., p. 1 –46, 2011.

KHAZANKIN, R.; DUSTDAR, S. On adaptive integration of web data sources into applications. In: 3RD INTERNATIONAL WORKSHOP ON INNOVATION IN INFORMATION TECHNOLOGIES - THEORY AND PRACTICE, DRESDEN, GERMANY, 1., 2010. **Anais**... [S.l.: s.n.], 2010, p. 16–19.

LIU, Z. H.; GAWLICK, D. Management of flexible schema data in rdbmss-opportunities and limitations for nosql-.. In: CIDR, .., 2015. **Anais**... [S.l.: s.n.], 2015. Disponível em: <http://cidrdb.org/cidr2015/program.html>. Acesso em: .

MEISEN, T.; REINHARD, R.; SCHILBERG, D. ; JESCHKE, S. A framework for adaptive data integration in digital production. In: JESCHKE, S.; ISENHARDT, I.; HEES, F. ; HENNING, K. (Org.). **Automation, Communication and Cybernetics in Science and Engineering 2011/2012**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 1053–1066. ISBN 978-3-642-33389-7.

MICHAEL, M.; MOREIRA, J. E.; SHILOACH, D. ; WISNIEWSKI, R. W. Scale-up x scale-out: A case study using nutch/lucene. In: 2007 IEEE INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM, 1., 2007. **Anais**... [S.l.: s.n.], 2007, p. 1–8.

PENTAHO. Data Integration - Kettle. Disponível em: <http://community.pentaho.com/projects/data-integration>. Acesso em: 23 outubro de 2016.

POESS, M.; RABL, T.; JACOBSEN, H.-A. ; CAUFIELD, B. Tpc-di: The first industry benchmark for data integration. **Proc. VLDB Endow.**, v. 7, n. 13, p. 1367–1378, 2014. Disponível em: <http://dx.doi.org/10.14778/2733004.2733009>. Acesso em: .

RED HAT, INC. Teiid - The data you want from the data you have. Disponível em: <http://teiid.jboss.org>. Acesso em: 23 outubro de 2016.

RED HAT, INC. Teiid Designer. Disponível em: <http://teiiddesigner.jboss.org/designer$_s ummary.html > .Acessoem$: 23$outubrode$2016.

RUSSOM, P. Data integration architecture. **What Works in Data integration Volume 25**, v. 25, 2008. Disponível em: <https://tdwi.org/articles/2008/05/27/data-integration-architecture-what-it-does-where-its-going-and-why-you-should-care.aspx>. Acesso em: Fri, 19 Feb 2016 20:04:00 -0200.

SADALAGE, P. J.; FOWLER, M. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**. 1st. ed. [S.l.]: Addison-Wesley Professional, 2012. ISBN 0321826620, 9780321826626.

SHETH, A. P. Changing focus on interoperability in information systems:from system, syntax, structure to semantics. In: GOODCHILD, M.; EGENHOFER, M.; FEGEAS, R. ; KOTTMAN, C. (Org.). **Interoperating Geographic Information Systems**. Boston, MA: Springer US, 1999. p. 5–29. ISBN 978-1-4615-5189-8.

SHETH, A. P.; LARSON, J. A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. **ACM Comput. Surv.**, v. 22, n. 3, p. 183–236, 1990. Disponível em: <http://doi.acm.org/10.1145/96602.96604>. Acesso em: Fri, 19 Feb 2016 20:04:00 -0200.

STONEBRAKER, M.; BRUCKNER, D.; ILYAS, I. F.; BESKALES, G.; CHERNIACK, M.; ZDONIK, S. B.; PAGAN, A. ; XU, S. Data curation at scale: The

data tamer system. In: CIDR 2013, SIXTH BIENNIAL CONFERENCE ON INNOVATIVE DATA SYSTEMS RESEARCH, ASILOMAR, CA, USA, JANUARY 6-9, 2013, ONLINE PROCEEDINGS, .., 2013. **Anais**... [S.l.: s.n.], 2013. Disponível em: <http://www.cidrdb.org/cidr2013/Papers/CIDR13$_{P}aper28.pdf$ > $.Acessoem$: $Fri, 19Feb201620 : 04 : 00 - 0200$.

STONEBRAKER, M.; CATTELL, R. 10 rules for scalable performance in 'simple operation' datastores. **Commun. ACM**, v. 54, n. 6, p. 72–80, 2011. Disponível em: <http://doi.acm.org/10.1145/1953122.1953144>. Acesso em: Fri, 19 Feb 2016 20:04:00 -0200.

STRNAD, P.; MACEK, O. ; JIRA, P. Mapping xml to key-value database. **English. In: DBKDA**, v. 1, p. 121–127, 2013.

WAZLAWICK, R. S. **Metodologia De Pesquisa Para Ciência Da Computação**. [S.l.]: Elsevier Editora Ltda, 2009. I - III p. ISBN 978-85-352-3522-7.

COLIN WHITE. Data Integration: Using Extract Transform and Load, Enterprise Architecture InitiativesEAI, and Enterprise Information Integration Tools to Create an Integrated Enterprise. Disponível em: <http://tdwi.org/articles/2006/05/09/data-integration-using-etl-eai-and-eii-tools-to-create-an-integrated-enterprise-report-excerpt.aspx>. Acesso em: 2015-05-15.

ZENTGRAF, R. **Estatística Objetiva**. Rio de Janeiro, RJ, Brasil: ZTG Editora LTDSA, 2001. ISBN 85-88309-01-7.

8 APÊNDICES

Especificação de Projeto de Sistemas

Flexible Data Integration - FlexDI

1 Diagrama de Caso de Uso

1.1 Diagrama de Pacotes - Casos de Uso

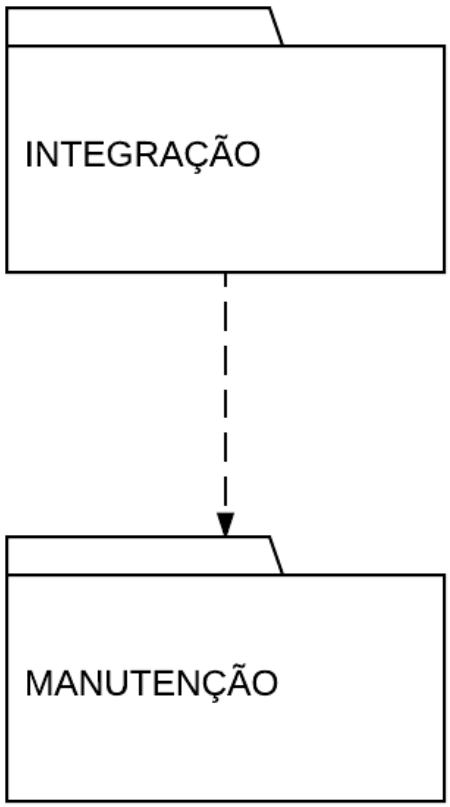

Figure 1: Diagrama de Pacotes - Casos de Uso

1.2 Casos de Uso - Integração

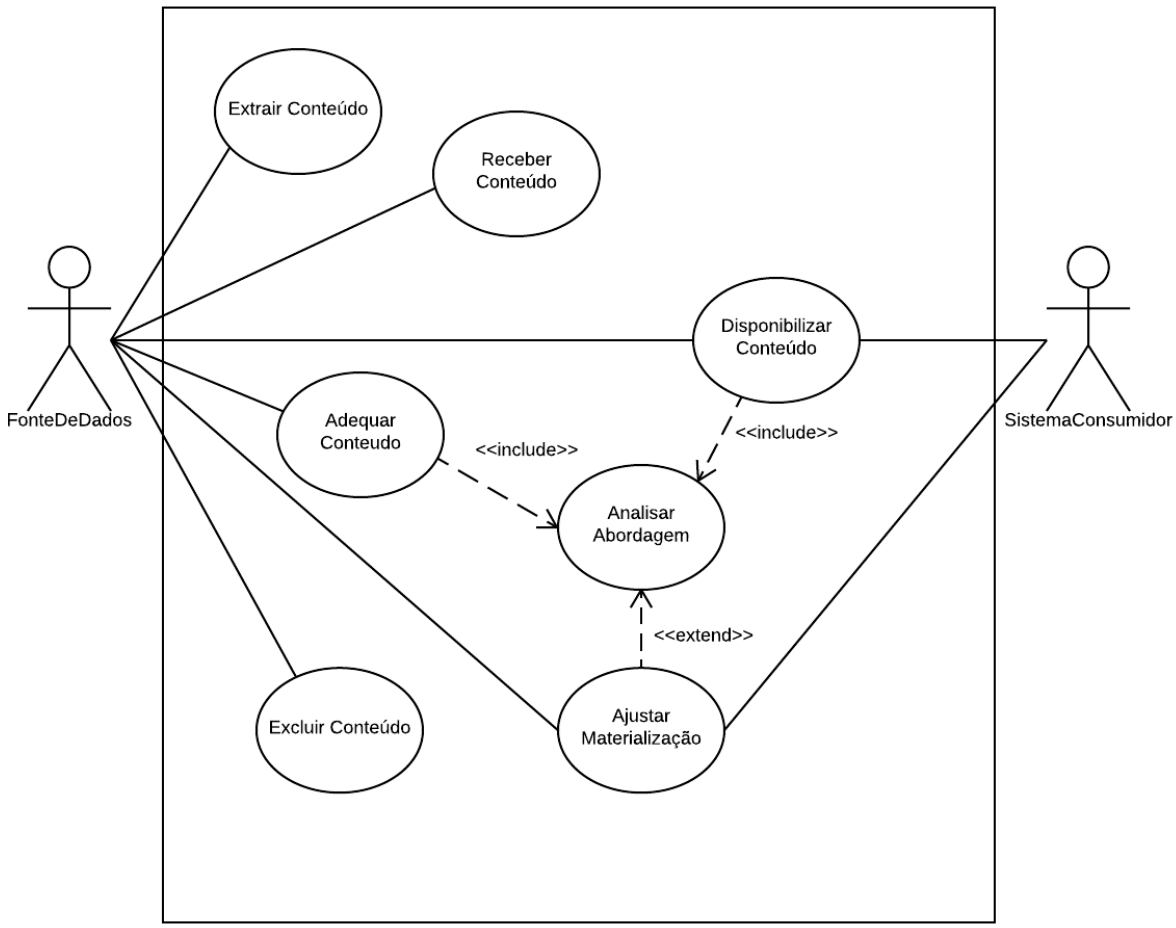

Figure 2: Casos de Uso - Integração

1.3 Casos de Uso - Manutenção

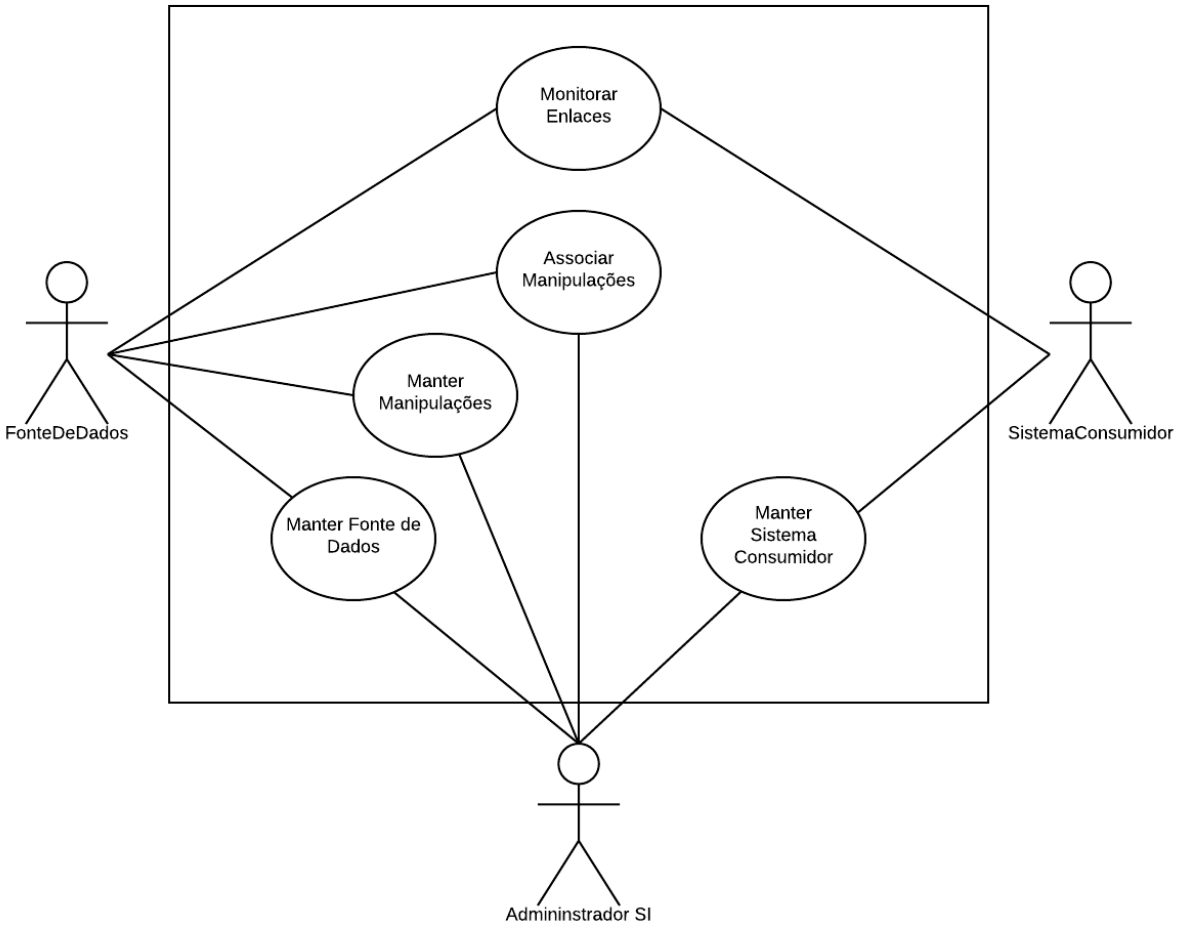

Figure 3: Casos de Uso - Manutenção

2 Descrição de Casos de Uso

2.1 Extrair Conteúdo

Objetivo Este caso de uso tem como objetivo descrever o processo de extração de um conteúdo de uma fonte de dados de comportamento passivo com abordagem de integração configurada como materialização

Atores: Fonte de Dados

Pré Condições: Este caso se inicia quando o evento de extração é acionado pela frequência de verificação de novos conteúdos do sistema

Fluxo Principal

1. O sistema verifica que há conectividade com a fonte de dados;
2. O sistema resgata os identificadores dos conteúdos extraídos da fonte de dados;
3. O sistema resgata os conteúdos da fonte de dados e calcula seus respectivos identificadores;
4. O sistema verifica que há identificadores calculados dos conteúdos que não existem na lista de identificadores resgatados dos conteúdos extraídos da fonte de dados;
5. O sistema extrai os conteúdos da fonte de dados que não estão na lista de identificadores resgatados dos conteúdos extraídos;
6. O sistema guarda a carga útil extraída dos conteúdos da fonte de dados, a data e a hora de sua guarda, seu volume e o identificador calculado;
7. O caso de uso termina com sucesso;

Fluxos Alternativos

A. Passo 1 do Fluxo Principal - Não há conectividade com a fonte de dados;

A.1. O sistema informa ao administrador que a fonte de dados não está acessível;

A.2. O caso de uso termina com falha.

B. Passo 2 do Fluxo Principal - Não há novo conteúdo na fonte de dados para ser extraído.

B.1. O caso de uso termina com sucesso.

C. Passo 3 do Fluxo Principal - Há identificadores calculados que estão na lista de identificadores resgatados

C.1. O sistema salva a data e hora atual como a data de ultima visita.

C.2. O caso de uso termina com sucesso.

Pós Condições

Sucesso : Novo conteúdo da fonte de dados extraído ou não há conteúdo novo a ser extraído;

Falha : Não há conectividade com a fonte de dados;

Outras Informações

– Calculo de identificador: função de hash (MD5) sobre a carga útil do conteúdo

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.2 Receber Conteúdo

Objetivo Este caso de uso tem como objetivo descrever o processo de recebimento de um conteúdo de uma fonte de dados de comportamento ativo

Atores: Fonte de Dados

Pré Condições: Este caso se inicia quando o sistema verifica que um conteúdo foi recebido da fonte de dados;

Fluxo Principal

1. O sistema calcula o identificador do conteúdo;
2. O sistema resgata os identificadores dos conteúdos recebidos;
3. O sistema verifica que o conteúdo enviado possui um identificador diferente daqueles resgatados dos conteúdos recebidos;
4. O sistema guarda carga útil do conteúdo recebido, a data e hora de seu recebimento, seu volume e o identificador calculado;
5. O caso de uso termina com sucesso;

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - Não há identificador diferente daqueles resgatados dos conteúdos enviados.

A.1. O sistema remove o conteúdo recebido;

A.2. O caso de uso termina com sucesso.

Pós Condições

Sucesso : Novo conteúdo da fonte de dados recebido ou não há conteúdo novo a ser materializado;

Falha : Não há conectividade com a fonte de dados;

Outras Informações

– Calculo de identificador: função de hash (MD5) sobre a carga útil do conteúdo

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.3 Excluir Conteúdo

Objetivo Este caso de uso tem como objetivo descrever o processo de remoção periódica dos conteúdos extraídos e integrados.

Atores: Fonte de Dados

Pré Condições: Este caso se inicia ao término do caso de uso *Ajustar Materialização*.

Fluxo Principal

1. O sistema resgata o tempo de vida do conteúdo manipulado;
2. O sistema verifica que a data e hora de guarda do conteúdo manipulado da fonte de dados é superior ou igual a data atual menos o tempo de vida do conteúdo manipulado;
3. O caso de uso termina com sucesso;

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - O sistema verifica que a data e hora de guarda do conteúdo manipulado é inferior a data atual menos o tempo de vida do conteúdo manipuladoss

A.1. O sistema remove os conteúdos extraídos e manipulados que tenham datas de guarda inferiores a data atual menos o tempo de vida do conteúdo manipulado;

A.2. O caso de uso termina com sucesso.

Pós Condições

– Conteúdos extraídos e manipulados que possuam datas de guarda inferiores a data atual menos o tempo de vida do conteúdo manipulado são removidas ou não há conteúdo extraídos ou manipulados a serem removidos.

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.4 Adequar Conteúdo

Objetivo Este caso de uso tem como objetivo descrever o processo de adequação do conteúdo extraído de uma fonte de dados com abordagem de integração configurada em materialização ou do conteúdo recebido de uma fonte de dados

Atores: Fonte de Dados

Pré Condições: Este caso se inicia quando o conteúdo extraído ou recebido é salvo no repositório de extração do sistema.

Fluxo Principal

1. O sistema recupera as manipulações pertinentes à materialização do conteúdo da fonte de dados;
2. Para cada manipulação de conteúdo relacionada em ordem crescente de aplicação
   2.1. O sistema aplica com sucesso a manipulação no conteúdo extraído;
3. O sistema guarda a carga útil do conteúdo manipulado, a data e a hora de guarda, o volume manipulado, o tempo gasto para adequá-lo e o identificador do conteúdo extraído;
4. *Executar Analisar Abordagem*;
5. O caso de uso termina com sucesso;

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - Há erro na aplicação da manipulação de conteúdo.

A.1. O sistema informa ao administrador que houve erro na aplicação da manipulação do conteúdo extraído, indicando o seu motivo;

A.2. O caso de uso termina com falha.

Pós Condições

Sucesso : O conteúdo manipulado a partir do conteúdo extraído é salvo no sistema

Falha : Há erro de manipulação do conteúdo extraído que impede sua integração

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.5 Analisar Abordagem

Objetivo Este caso de uso tem como objetivo descrever o processo de seleção da abordagem de integração mais apropriada para uma determinada fonte de dados em um determinado momento do tempo de vida do ambiente de integração.

Atores: Fonte de Dados, Sistema Consumidor

Pré Condições: Este caso se inicia ao término da adequação de um conteúdo ou ao término do consumo de um conteúdo por um sistema consumidor.

Fluxo Principal

1. O sistema recupera o comportamento e a capacidade de resposta da fonte de dados assim como a existência de sintaxe e modelo lógico do conteúdo da fonte de dados;
2. O sistema verifica que o comportamento e a capacidade de resposta da fonte de dados, a existência de sintaxe e modelo lógico no conteúdo permitem a virtualização da fonte de dados;
3. O sistema recupera o tempo de vida e da frequência de atualização do conteúdo na fonte dados;
4. O sistema recupera o tempo total de processamento para adequação do conteúdo da fonte de dados;
5. O sistema recupera o tempo de transporte do conteúdo da fonte dados até sistema consumidor;
6. O sistema recupera a probabilidade de perda admitida pelo sistema consumidor para a fonte de dados em questão;
7. O sistema verifica que o intervalo de atualização do conteúdo na fonte de dados é inferior ou igual ao tempo de vida do conteúdo na fonte de dados subtraído do tempo de transporte do conteúdo da fonte de dados até o sistema consumidor e do tempo total de processamento do conteúdo ;
8. O sistema verifica que a frequência de verificação do sistema consumidor é igual ou superior a frequência de atualização do conteúdo na fonte de dados ;
9. O sistema verifica que a abordagem de integração utilizada pela fonte de dados é a virtualização;
10. O caso de uso termina com sucesso.

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - Ou o comportamento do invólucro, ou a capacidade de resposta do invólucro, ou a existência de sintaxe no conteúdo ou a existência de modelo lógico não permitem a virtualização

A.1. O sistema estabelece que a abordagem de integração adequada é a materialização;

A.2. *Executar Ajustar Materialização*;

A.3. O caso de uso termina com sucesso.

B. Passo 6 do Fluxo Principal - O sistema verifica que o intervalo de atualização do conteúdo na fonte de dados é superior ao tempo de vida do conteúdo na fonte de dados subtraído do tempo de transporte do conteúdo da fonte de dados para o sistema consumidor e do tempo total de processamento do conteúdo no sistema;

B.1. O sistema calcula a frequência ideal de verificação de novos conteúdos pelo sistema consumidor;[ref formula]

B.2. O sistema verifica que atual frequência de verificação do sistema consumidor é igual ou superior à frequência ideal calculada;

B.3. O sistema verifica que a abordagem de integração utilizada pela fonte de dados é a virtualização;

B.4. O caso de uso termina com sucesso.

C. Passo B.3. do Fluxo Alternativo B. - A frequência de verificação de novo conteúdos pelo sistema consumidor é inferior à frequência ideal calculada;

C.1. O sistema ajusta a a abordagem de integração da fonte de dados para materialização;

C.2. O sistema habilita a verificação de novos conteúdos para materialização;

C.3. *Executar Ajustar Materialização*

C.4. O caso de uso termina com sucesso.

D. Passo 9 do Fluxo Principal e Passo B.3. do Fluxo Alternativo B. - O sistema verifica que a abordagem de integração utilizada pela fonte de dados é a materialização

D.1. O sistema ajusta a a abordagem de integração da fonte de dados para virtualização;

D.2. O sistema desabilita a verificação de novos conteúdos para materialização;

D.3. O caso de uso termina com sucesso.

Pós Condições

- A abordagem de integração da fonte de dados é configurada para virtualização uma vez que as características estáticas permitem e a frequência de verificação de novos conteúdos pelo sistema consumidor é igual ou superior a frequência de atualização da fonte de dados;
- A abordagem de integração da fonte de dados é configurada para materialização uma vez que as características estáticas não permitem ou a frequência de verificação de novos conteúdos pelo sistema consumidor é inferior a frequência de atualização da fonte de dados;

Outras Informações

– Cálculo da frequência ideal

$$n_{SC} = \begin{cases} 1 & , \quad t^{'} t_a^{FD} \\ \lceil \dfrac{\ln(p(x=0)^{SC})}{\ln(1 - t^{'} * f_a^{FD})} \rceil & , \quad t^{'} < t_a^{FD} \end{cases} \tag{1}$$

$$t^{'} = t_v^{FD} - t_T^{FDSI} - t_T^{SISC} - t_p$$

$$f_V^{SC} n_{SC} * f_a^{FD} \tag{2}$$

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.6 Ajustar Materialização

Objetivo Este caso de uso tem como objetivo descrever o processo de ajuste da frequência de verificação de novos conteúdos pelo sistema em uma determinada fonte de dados que possui a materialização como sua abordagem de integração.

Atores: Fonte de Dados, Sistema Consumidor

Pré Condições: Este caso se inicia após o consumo de um conteúdo de uma fonte de dados que possui a materialização como sua abordagem de integração ou quando a abordagem de integração passa de virtualização para materialização após a análise de abordagem

Fluxo Principal

1. O sistema recupera o tempo de vida e a frequência de atualização do conteúdo da fonte dados;
2. O sistema recupera o tempo de transporte do conteúdo disponível para o repositório de conteúdos extraídos;
3. O sistema verifica que o tempo de vida do conteúdo na fonte de dados, subtraído do tempo de transporte do conteúdo para o repositório de conteúdos extraídos, é superior ou igual ao intervalo de de atualização do conteúdo na fonte de dados;
4. O sistema ajusta a frequência de verificação da fonte de dados para um valor igual ou superior à frequência de atualização do conteúdo disponível na fonte de dados
5. O sistema recupera o tempo de vida e a frequência de atualização do conteúdo manipulado;
6. O sistema recupera o tempo de transporte do conteúdo manipulado para o sistema consumidor;
7. O sistema verifica que o tempo de vida do conteúdo manipulado, subtraído do tempo de transporte do conteúdo manipulado para o sistema consumidor, é superior ao intervalo de atualização do conteúdo na fonte de dados.
8. O sistema verifica que a frequência de verificação do sistema consumidor é igual ou superior à frequência de atualização do conteúdos manipulado;
9. O caso de uso termina com sucesso.

Fluxos Alternativos

A. Passo o Fluxo Principal - O sistema verifica que o tempo de vida do conteúdo na fonte de dados, subtraído do tempo de transporte do conteúdo para o repositório de conteúdos extraídos, é inferior ao intervalo de de atualização do conteúdo na fonte de dados.

A.1. O sistema recupera a probabilidade de perda de conteúdo extraído;

A.2. O sistema calcula a frequencia ideal de verificação de novos conteúdos;

A.3. O sistema verifica que é possível ajustar a frequência de verificação de novos conteúdos ao valor calculado como ideal;

A.4. O sistema ajusta a frequência de verificação para o valor de frequência ideal calculado;

A.5. Retorna ao passo 6 do Fluxo Principal.

B. Passo 7 do Fluxo Principal - O tempo de vida do conteúdo manipulado, subtraído do tempo de transporte do conteúdo manipulado ao sistema consumidor, é inferior ao intervalo de de atualização do conteúdo manipulado.

B.1. O sistema recupera a probabilidade de perda admitida pelo sistema consumidor;

B.2. O sistema calcula a frequencia ideal de verificação de novos conteúdos pelo sistema consumidor;

B.3. O sistema verifica que é a frequência de verificação de novos conteúdos pelo sistema consumidor é superior ou igual ao valor ideal calculado;

B.4. O caso de uso termina com sucesso.

C. Passo A.4. do Fluxo Alternativo A. A - O sistema não é capaz de ajustar a frequência de verificação de conteúdo disponível

C.1. O sistema informa ao administrador do sistema que não é possível ajusta a frequência de verificação de novos conteúdos para o valor calculado;

C.2. O caso de uso termina com falha.

D. Passo B.3.do Fluxo Alternativo B. - O sistema consumidor possui uma frequência de verificação inferior a ideal;

D.1. O sistema informa ao administrador do sistema que a frequência de verificação do sistema consumidor está abaixo do ideal;

D.2. O caso de uso termina com falha.

Pós Condições

Sucesso : O sistema é capaz de ajustar a frequência de verificação de novos conteúdos ou o sistema consumidor possui uma frequência de verificação adequada.

Falha : O sistema não é capaz de ajustar a frequência de verificação de novos conteúdos ou o sistema consumidor nãopossui uma frequência de verificação adequada.

Outras Informações

– Cálculo da frequência ideal

$$n_{SI} = \begin{cases} 1 & , \quad (t_v^{FD} - t_T^{FDSI}) t_a^{FD} \\ \lceil \dfrac{\ln(p(x=0)^{SI})}{\ln(1-(t_v^{FD} - t_T^{FDSI}) * f_a^{FD})} \rceil & , \quad (t_v^{FD} - t_T^{FDSI}) < t_a^{FD} \end{cases} \tag{3}$$

$$f_V^{SI} n_{SI} * f_a^{FD} \tag{4}$$

$$n_{SC} = \begin{cases} 1 & , \quad (t_v^{SI} - t_T^{SISC}) t_a^{SI} \\ \lceil \dfrac{\ln(p(x=0)^{SC})}{\ln(1-(t_v^{SI} - t_T^{SISC}) * f_a^{SI})} \rceil & , \quad (t_v^{SI} - t_T^{SISC}) < t_a^{SI} \end{cases} \tag{5}$$

$$f_V^{SC} n_{SC} * f_a^{SI} \tag{6}$$

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.7 Disponibilizar Conteúdo

Objetivo Este caso de uso tem como objetivo descrever o processo de consumo de conteúdos de uma fonte de dados por um sistema consumidor.

Atores: Sistema Consumidor, Fonte de Dados

Pré Condições: Este caso se inicia quando o sistema consumidor solicita o conteúdo de uma fonte de dados

Fluxo Principal

1. O sistema registra a data e a hora que o sistema consumidor solicita o conteúdo da fonte de dados;
2. O sistema verifica que a conexão com o repositório do conteúdo manipulado está disponível;
3. O sistema verifica que a conexão com o repositório com a fonte de dados está disponível;
4. O sistema resgata e une com sucesso o conteúdo da fonte de dados e o conteúdo manipulado relativos à solicitação pelo sistema consumidor;
5. O sistema aplica com sucesso as manipulações na união do conteúdo da fonte de dados e o conteúdo manipulado em ordem crescente de aplicação;
6. O sistema calcula o volume do conteúdo processado;
7. O sistema guarda a data, a hora, o tempo gasto para aplicar as manipulações de conteúdo e o volume de conteúdo processado
8. O sistema disponibiliza o conteúdo processado ao sistema consumidor;
9. *Executar Analisar Abordagem*;
10. O caso de uso termina com sucesso.

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - Não há conectividade com o conteúdo manipulado

A.1. O sistema informa ao administrador que o conteúdo manipulado não está acessível;

A.2. O caso de uso termina com falha.

B. Passo 3 do Fluxo Principal - Não há conectividade com o conteúdo da fonte de dados

B.1. O sistema informa ao administrador que o conteúdo da fonte de dados não está acessível;

B.2. O caso de uso termina com falha.

C. Passo 4 do Fluxo Principal - Há erro na união do conteúdo manipulado e do conteúdo da fonte de dados.

C.1. O sistema informa ao administrador que houve erro na união do conteúdo manipulado e do conteúdo da fonte de dados solicitada;

C.2. O caso de uso termina com falha.

D. Passo 5 do Fluxo Principal - Há erro na aplicação das manipulações de conteúdo.

D.1. O sistema informa ao administrador que houve erro na aplicação das manipulações de conteúdo na união do conteúdo da fonte de dados e o conteúdo manipulado, indicando em qual ocorreu o erro e a especificação deste erro;

D.2. O caso de uso termina com falha.

Pós Condições

Sucesso : O conteúdo é consumido com sucesso

Falha : O conteúdo solicitado pelo sistema consumidor não pode ser entregue pelo sistema

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.8 Manter Fonte de Dados

Objetivo Este caso tem como objetivo descrever as funcionalidades de manutenção do cadastro de fontes de dados

Atores: Fonte de Dados

Pré Condições: Este caso se inicia quando o administrador da solução de integração acessa a área de cadastro das fontes de dados

Fluxo Principal

1. O sistema apresenta as opções de Incluir, Consultar, Modificar e Excluir uma fonte de dados ;
2. O administrador do sistema seleciona Incluir;
3. O sistema solicita o nome da fonte de dados,
4. O administrador do sistema preenche o nome da fonte de dados;
5. O sistema solicita o comando de conexão com o invólucro da fonte de dados;
6. O administrador do sistema preenche o comando de conexão com o invólucro;
7. O sistema solicita o comportamento da fonte e a capacidade de consulta do invólucro da fonte de dados;
8. O administrador do sistema preenche o comportamento da fonte e a capacidade de consulta;
9. O sistema solicita solicita se existe sintaxe e modelo lógico reconhecido no conteúdo;
10. O administrador do sistema preenche sobre a existência de sintaxe e de modelo lógico do conteúdo;
11. O sistema solicita o tempo de vida e a cadência de atualização esperada do conteúdo em seu repositório original ;
12. O administrador do sistema preenche tempo de vida e a cadência de atualização esperada do conteúdo em seu repositório original ;
13. O sistema solicita o tempo de vida e a cadência de atualização esperada do conteúdo no repositório de integração ;
14. O administrador do sistema preenche tempo de vida e a cadência de atualização esperada do conteúdo no repositório de integração;
15. O sistema habilita o temporizador de verificação de novos conteúdos das fonte de dados e o de remoção de conteúdos integrados no repositório de integração.
16. O sistema verifica que existem condições para abordagem flexível e configura o atributo correspondente como verdadeiro;
17. O sistema configura a abordagem de integração da fonte de dados como de materialização.

18. O caso de uso termina com sucesso

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - O administrador do sistema seleciona Consultar

A.1. O sistema listas as fonte de dados cadastradas e a atual abordagem de integração adotada;

A.2. O caso de uso termina com sucesso.

B. Passo 2 do Fluxo Principal - O administrador do sistema seleciona Modificar

B.1. O sistema listas as fonte de dados cadastradas e a atual abordagem de integração adotada;

B.2. O administrador do sistema seleciona uma fonte de dados;

B.3. O sistema retorna os atributos estáticos e dinâmicos da fonte de dados;

B.4. O administrador do sistema altera os valores pertinentes ;

B.5. O sistema guarda os valores alterados;

B.6. Retorna para o passo 16 do Fluxo Principal

C. Passo 2 do Fluxo Principal - O administrador do sistema seleciona Excluir

C.1. O sistema listas as fonte de dados cadastradas e a atual abordagem de integração adotada;

C.2. O administrador do sistema seleciona uma fonte de dados para excluir;

C.3. O sistema exclui a fonte de dados indicada

C.4. O caso de uso termina com sucesso.

D. Passo 16 do Fluxo Principal - Não há condições de utilizar uma abordagem de integração flexível

D.1. O sistema configura o atributo correspondente como falso;

D.2. Retorna para o passo 17 do Fluxo Principal

Pós Condições

– Não há

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.9 Manter Manipulações

Objetivo Este caso tem como objetivo descrever as funcionalidades de manutenção da manipulação do conteúdo das fontes de dados

Atores: Fonte de Dados

Pré Condições: Este caso se inicia quando o administrador da solução de integração acessa a área de cadastro das manipulações.

Fluxo Principal

1. O sistema apresenta as opções de Incluir, Consultar, Modificar e Excluir uma manipulação;
2. O administrador do sistema seleciona Incluir;
3. O sistema solicita o nome da manipulação e a ordem de aplicação no conteúdo da fonte de dados
4. O administrador do sistema preenche o nome da manipulação e sua ordem de aplicação;
5. O sistema solicita a carga da descrição da manipulação conforme padrão acordado;
6. O administrador do sistema carrega com sucesso a descrição da manipulação.
7. O caso de uso termina com sucesso

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - O administrador do sistema seleciona Consultar

A.1. O sistema lista as manipulações cadastradas da fonte de dado por ordem de aplicação;

A.2. O caso de uso termina com sucesso..

B. Passo 2 do Fluxo Principal - O administrador do sistema seleciona Modificar

B.1. O sistema lista as manipulações cadastradas da fonte de dado por ordem de aplicação;

B.2. O administrador do sistema seleciona uma manipulação;

B.3. O sistema solicita a nova carga da descrição da manipulação;

B.4. O administrador do sistema carrega com sucesso a nova descrição da manipulação.

B.5. O sistema guarda a manipulação carregada;

B.6. O caso de uso termina com sucesso.

C. Passo 2 do Fluxo Principal - O administrador do sistema seleciona Excluir

C.1. O sistema lista as manipulações cadastradas da fonte de dado por ordem de aplicação;

C.2. O administrador do sistema seleciona uma manipulação para excluir;

C.3. O sistema exclui a manipulação indicada e reoordena a ordem de aplicação;

C.4. O caso de uso termina com sucesso.

Pós Condições

– Não há

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.10 Monitorar Enlaces

Objetivo Este caso de uso tem como objetivo descrever o processo de monitoração dos enlaces das fontes de dados e dos sistemas consumidores com o sistema

Atores: Fonte de Dados, Sistema Consumidor

Pré Condições: Este caso se inicia quando o evento de monitoração da latência dos enlaces é acionado

Fluxo Principal

1. O sistema recupera o endereço de conexão das fontes de dados
2. Para cada endereço de conexão da recuperado das fontes de dados
   2.1. O sistema verifica que há conectividade com a fonte de dados;
   2.2. O sistema dispara o comando de medição de taxa de transmissão para o endereço de conexão da fonte de dados;
   2.3. O sistema recebe a medição de taxa de transmissão
   2.4. O sistema guarda a data e hora de medição e a taxa de transmissão medida.
3. O sistema recupera o endereço de conexão dos sistemas consumidores
4. Para cada endereço de conexão recuperado dos sistemas consumidores
   4.1. O sistema verifica que há conectividade com o sistema consumidor;
   4.2. O sistema dispara o comando de medição de taxa de transmissão para o endereço de conexão do sistema consumidor;
   4.3. O sistema operacional retorna a medição de taxa de transmissão
   4.4. O sistema guarda a data e hora de medição e a taxa de transmissão medida.
5. O caso de uso termina com sucesso.

Fluxos Alternativos

Pós Condições

Sucesso :

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.11 Associar Manipulação

Objetivo Este caso de uso tem como objetivo descrever o processo de associação de manipulações a um conteúdo de uma fonte de dados

Atores: Fonte de Dados

Pré Condições: O caso de uso de inicia com a seleção de uma fonte de dados pelo administrador da solução de integração.

Fluxo Principal

1. O sistema resgata as manipulações de conteúdo cadastradas e apresenta ao administrador da solução de integração;
2. O administrador da solução seleciona uma manipulação da lista de manipulações
3. O sistema solicita a que tipo de abordagem de integração se aplica a a manipulação selecionada
4. O administrador seleciona que a manipulação a ser aplicada é comum às abordagens de integração.
5. O sistema solicita a ordem de aplicação da manipulação;
6. O administrador indica a ordem em que a manipulação será aplicada.
7. O sistema guarda as informações dadas pelo administrador e termina o caso com suceso.

Fluxos Alternativos

A. Passo 4 do Fluxo Principal - O administrador seleciona que a manipulação a ser aplicada é exclusiva à abordagens de integração de materialização.

A.1. Retorna ao passo 5 do Fluxo Principal.

Pós Condições

Sucesso :

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

2.12 Manter Sistemas Consumidores

Objetivo Objetivo : Este caso tem como objetivo descrever as funcionalidades de manutenção do cadastro de fontes de dados

Atores: Sistema Consumidor

Pré Condições: Este caso se inicia quando o administrador da solução de integração acessa a área de cadastro dos sistemas consumidores.

Fluxo Principal

1. O sistema apresenta as opções de Incluir, Consultar, Modificar e Excluir uma fonte de dados ;
2. O administrador do sistema seleciona Incluir ;
3. O sistema solicita o nome do sistema consumidor,
4. O administrador do sistema preenche o nome do sistema consumidor;
5. O sistema solicita o comando de conexão com a fonte de dados;
6. O administrador do sistema preenche o comando de conexão;
7. O sistema solicita a probabilidade de perda admitida pelo sistema consumidor;
8. O administrador do sistema preenche a probabilidade de perda admitida pelo sistema consumidor;
9. O caso de uso termina com sucesso

Fluxos Alternativos

A. Passo 2 do Fluxo Principal - O administrador do sistema seleciona Consultar

A.1. O sistema listas os sistemas consumidores cadastrados e a probabilidade de perda de configurada para cada um;

A.2. O caso de uso termina com sucesso.

B. Passo 3 do Fluxo Principal - O administrador do sistema seleciona Modificar

B.1. O sistema listas os sistemas consumidores cadastrados;

B.2. O administrador do sistema seleciona um sistema consumidor;

B.3. O sistema retorna o comando de conexão e a probabilidade de perda admitida;

B.4. O administrador do sistema altera os valores pertinentes ;

B.5. O sistema guarda os valores alterados;

B.6. O caso de uso termina com sucesso.

C. Passo 3 do Fluxo Principal - O administrador do sistema seleciona Excluir

C.1. O sistema listas os sistemas consumidores cadastrados;

C.2. O administrador do sistema seleciona um sistema consumidor para excluir;

C.3. O sistema exclui a fonte de dados indicada

C.4. O caso de uso termina com sucesso.

Pós Condições

Sucesso :

Outras Informações

– Não há

Pontos de Extensão

– Não há

Requisitos Não Funcionais

– Não definidos

3 Diagrama de Classe Preliminar

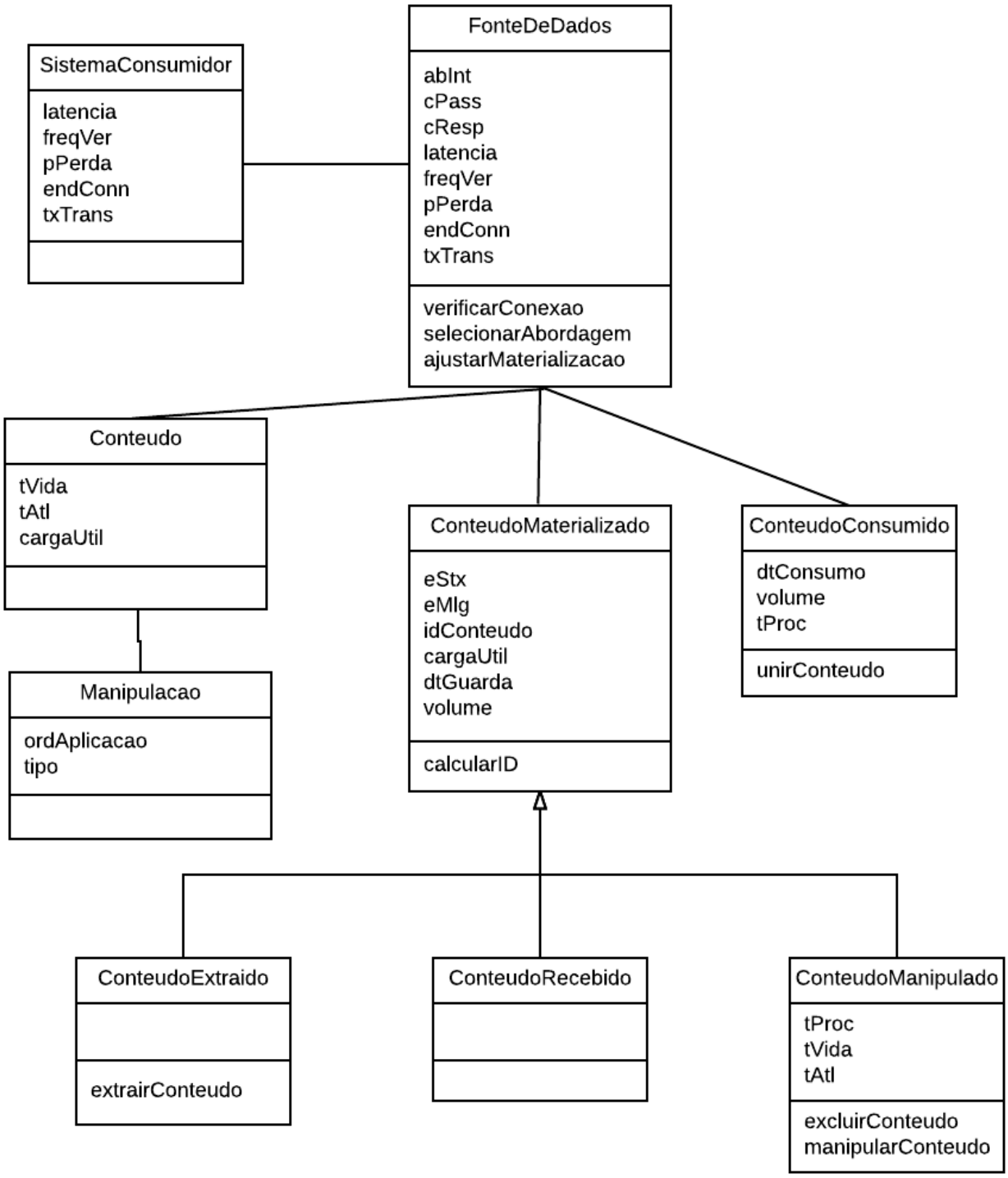

Figure 4: Diagrama de Classe Preliminar