DESENVOLVIMENTO E AVALIAÇÃO DE FERRAMENTAS COMPUTACIONAIS PARA TRIAGEM AUTOMÁTICA DE SUJEITOS DE PESQUISA

DIOGO FERREIRA DA COSTA PATRÃO

Tese apresentada à Fundação Antônio Prudente para obtenção do Título de Doutor em Ciências

Área de concentração: Oncologia

Orientadora: Profª. Drª. Helena Brentani
Co-Orientadora: Profª. Drª. Renata Wassermann

São Paulo
2014

FICHA CATALOGRÁFICA

Preparada pela Biblioteca da Fundação Antônio Prudente

da Costa Patrão, Diogo Ferreira

Desenvolvimento e avaliação de ferramentas computacionais para triagem automática de sujeitos de pesquisa / Diogo Ferreira da Costa Patrão - São Paulo, 2014.

94p.

Tese (Doutorado) - Fundação Antônio Prudente.

Curso de Pós-Graduação em Ciências - Área de Concentração: Oncologia

Orientadora: Helena Brentani

Descritores: 1. SELEÇÃO DE PACIENTES. 2. REPRESENTAÇÃO DE CONHECIMENTO (COMPUTADOR). 3. INTELIGÊNCIA ARTIFICIAL 4. INTEGRAÇÃO DE SISTEMAS. 5. PRONTUÁRIO ELETRÔNICO

A verdade é a mentira que aconteceu.

Luis Fernando Verissimo, “O analista de Bagé”

DEDICATÓRIA

Para Franklin Ferreira Leite.

RESUMO

da Costa Patrão DF. **Desenvolvimento e avaliação de ferramentas computacionais para triagem automática de sujeitos de pesquisa**. São Paulo; 2014. [Tese de Doutorado - Fundação Antônio Prudente].

O sucesso de um projeto de pesquisa em medicina depende do recrutamento de um número suficiente de participantes de pesquisa. Um dos desafios para atingir a quantidade adequada é como utilizar dados de prontuário eletrônico para acelerar a avaliação de pacientes. Atualmente, isto é feito por revisão manual dos prontuários, um processo demorado e propenso a erros. Neste trabalho, especificamos e implementamos Ontocloud, um sistema de integração de dados baseado reescrita de consultas e em ontologias, com capacidade de inferência, para seleção de pacientes que atendam a critérios clínicos utilizando dados de prontuário eletrônico. Aplicamos este sistema a um estudo clínico real, conduzido no AC Camargo Cancer Center, e verificamos que atendeu a todos os critérios especificados e resolveu adequadamente o problema de seleção de participantes de pesquisa. Ainda, mostramos que sua performance é compatível com sistemas de integração similares.

SUMMARY

da Costa Patrão DF. **[Development and evaluation of computational tools for automatic selection of research subjects]** . São Paulo; 2014. [Tese de Doutorado - Fundação Antônio Prudente].

A successful medical research project is entirely dependent on enough subjects being recruited. Among the challenges to achieve recruitment target, using avaliable data from electronic medical records to speed up the patient identification process. In this thesis, we specified and implemented Ontocloud, a query-rewriting, inference capable, ontology based data integration system, for selection of patients meeting clinical criteria using electronic medical records data. We applied it to a real clinical trial conducted at the AC Camargo Cancer Center and verified that it fulfilled all specified requirements, effectively solving the research subject selection problem. Also, we showed that Ontocloud performance is compatible with similar data integration systems.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIAÇÕES

BDR	**B**anco de **D**ados **R**elacional
BER	**B**ayes **E**rror **R**ate (taxa de erro de Bayes)
CID-10	**C**lassificação **I**nternacional de **D**oenças, décima edição
CISH	**C**hromogenic **I**n **S**itu **H**ibridization (hibridação in situ cromogênica)
DL	**D**escription **L**ogic (Lógica de descrições)
ETL	**E**xtract, **T**ransform and **L**oad (Extração, transformação e carga)
FISH	**F**luorescence **I**n **S**itu **H**ibridization (hibridação in situ fluorescente)
GAV	**G**lobal **A**s **V**iew (Global como visão)
LAV	**L**ocal **A**s **V**iew (Local como visão)
OBDI	**O**ntology **B**ased **D**ata **I**ntegration (Integração de dados baseado em ontologias)
OBDA	**O**ntology **B**ased **D**ata **A**ccess (Acesso a dados baseado em ontologias)
OWL	**W**eb **O**ntology **L**anguage (Linguagem de ontologias web)
NCI	**N**ational **C**ancer **I**nstitute (Instituto Nacional de Câncer - EUA)
RDF	**R**esource **D**escription **F**ramework (Arcabouço para descrição de recursos)
RDFS	**R**esource **D**escription **F**ramework **Schema** (Arcabouço para descrição de recursos)
SNOMED	**S**ystematized **NO**menclature of **Medicine** (nomenclatura sistematizada de Medicina)

SNOMED-CT	**S**ystematized **NO**menclature of **Medicine** - **C**linical **Terms** (nomenclatura sistematizada de Medicina - termos clínicos)
SPARQL	**S**PARQL **P**rotocol and **R**DF **Q**uery **L**anguage (Linguagem e protocolo de consulta a RDF)
SQL	**S**tructured **Q**uery **L**anguage (Linguagem estruturada de consultas, utilizada em bancos de dados relacionais)
TNM	Sistema de classificação de tumores malignos (**T**umour, **N**odes, **M**ethastasis)
URI	**U**niversal **Resource I**dentifier (Identificador Universal de Recursos)
URL	**U**niform **R**esource **L**ocator

ÍNDICE

1 INTRODUÇÃO

Com a tendência mundial de universalização dos serviços de saúde, a preocupação com prevenção e especialização dos métodos de diagnóstico e tratamento, a procura por serviços de saúde tem aumentado. Para atender essa demanda, hospitais tem adotado uma abordagem na qual um mesmo paciente será atendido por muitos profissionais diferentes ao longo do diagnóstico, tratamento e acompanhamento de uma doença.

Essa abordagem, ao mesmo tempo que é louvável por disponibilizar e baratear o acesso à saúde, melhorando a qualidade de vida de camadas mais carentes da população, depende da adequada comunicação entre os profissionais da saúde responsáveis por esse paciente. O prontuário do paciente é hoje o principal instrumento de comunicação e registro de informações sobre saúde.

Paradoxalmente, a informatização do prontuário do paciente não traz necessariamente melhor qualidade e agilidade no acesso à informação clínica. Em muitos casos, como o de atendimento ambulatorial, o registro dessas informações não é feito em formulários estruturados, mas em textos descritivos, como uma transcrição do que seria escrito à mão num prontuário em papel. Além disso, com a crescente especialização e evolução dos serviços de saúde, equipamentos de diagnóstico e tratamentos, surgem diferentes sistemas de informação especializados. A informação de um mesmo paciente fica distribuída em diferentes bancos de dados, dificultando a realização de consultas.

Em Ciência da Computação, o problema de realizar consultas que especifiquem diversos bancos de dados é conhecido como *Integração de Dados*.

Há casos em que, em mais de um banco de dados, existem informações similares representadas de maneiras diferentes, sintática ou semanticamente. Para que seja possível consultar estas informações em um sistema de integração de dados, é preciso que as primeiras sejam *harmonizadas*, podendo haver perda de detalhamento (ou granularidade). O processo conhecido como *inferência* permite a extração de novas informações a partir de um conjunto de fatos e relações entre conceitos (ou *axiomas*). *Ontologias* permitem a representação de conhecimento e sua semântica define regras para inferência. Bancos de dados relacionais, que são a tecnologia predominante em sistemas de informação (inclusive hospitalares) atualmente, não são capazes de realização de inferência.

O prontuário em papel, onde uma cópia física de cada um dos documentos gerados no atendimento do paciente é armazenada, serve como um repositório unificado de todas as informações do paciente, não importando se foi criado à mão ou por um dos múltiplos sistemas de informação utilizados naquela instituição. Porém, os documentos em papel trazem problemas próprios de conservação, manipulação e em especial disponibilidade.

Para a realização de pesquisas científicas em medicina, o recrutamento de sujeitos de pesquisa que atendam aos critérios de inclusão e exclusão depende da avaliação de informações, muitas das quais constam do prontuário do paciente. Onde há a adoção de ferramentas de PEP (Prontuário Eletrônico do Paciente), o processo de recrutamento poderia ser acelerado por sistemas de busca informatizada, que indicam quais pacientes poderiam atender aos critérios, baseado nas informações cadastradas no sistema. Entretanto, esses critérios são bastante complexos e a informação capturada durante o atendimento de rotina de um paciente normalmente está em formato diferente do que seria necessário para realizar uma busca automatizada. Na prática, o recrutamento de pacientes ainda depende de esforço humano para ser realizado, e numa instituição que atende a milhares de pacientes diariamente, é

inviável verificar um a um todos os prontuários. Isso implica num recrutamento sub-ótimo, dificultando a realização de pesquisas de alto impacto científico.

1.1 ESTUDOS CLÍNICOS

"Estudo clínico" é a denominação dada a todo projeto de pesquisa que visa avaliar a eficiência de novas intervenções em comparação aos já estabelecidos. O Centro de Apoio a Estudos Patrocinados (CAEP) é um departamento do A.C. Camargo Cancer Center que tem como objetivo facilitar a execução de projetos de pesquisa patrocinados por empresas farmacêuticas. Esses projetos, uma vez aprovados no Comitê de Ética em Pesquisa (CEP) da instituição, têm como primeira etapa o recrutamento dos participantes: nessa, pacientes que se enquadrem nos critérios de inclusão e exclusão são identificados e, se elegíveis, a participação no projeto é oferecida pelos médicos participantes do estudo.

Os critérios de inclusão e exclusão, para esse tipo de estudo, são numerosos e delimitam uma população com características específicas, para a qual a droga em avaliação é visada. Para avaliar quais pacientes da instituição têm potencial para inclusão no estudo, alguns dos critérios são avaliados tendo como base as informações constantes no prontuário do paciente. Nem todos os critérios podem ser verificados dessa maneira, pois os estudos por vezes exigem exames que não são rotineiros para aquele tipo de paciente. Assim, os critérios são separados em dois grupos, os de pré-triagem (que podem ser verificados no prontuário) e os de triagem (que exigem exames específicos para aquele estudo). Os pacientes que atendam aos critérios de pré-triagem são elencados para a triagem, no qual alguns exames mais específicos podem ser feitos. Aos pacientes que atenderem aos critérios de triagem após os exames, é oferecida a possibilidade de participar do estudo.

Em particular, alguns critérios de exclusão especificam um período de

tempo máximo após algum evento (como diagnóstico, ou final do último tratamento), ou tratamentos. Isso implica que é necessário identificar rapidamente os potenciais participantes de pesquisa, sob pena de não mais atenderem aos critérios. De uma forma geral, atingir o número necessário de participantes de pesquisa é um desafio enfrentado por pesquisadores no mundo inteiro - aproximadamente 50% dos estudos clínicos atingem sua meta de recrutamento, e desses, apenas 50% o faz dentro do prazo estabelecido. Apesar de haver publicações abordando soluções desenvolvidas para projetos de pesquisa específicos, a maioria se utiliza de soluções de integração já construídas com outros propósitos, ou não descreve como resolver os problemas de integração que são específicos deste tipo de informação.

Nesta tese, especificamos um sistema para seleção de potenciais participantes de pesquisa para projetos de pesquisa médicos, baseado em dados de prontuário eletrônico. O principal diferencial desse sistema é sua capacidade de realização de inferências sobre dados contidos em diversos bancos de dados relacionais. Avaliaremos sua performance em relação outras especificações de sistema com abordagens alternativas, aplicando-o a um estudo específico desenvolvido no A.C. Camargo Cancer.

2 OBJETIVO

O objetivo deste trabalho é especificar, implementar e avaliar um sistema de seleção de participantes de pesquisa baseado em dados de prontuário eletrônico.

2.1 CONTRIBUIÇÕES

Identificamos e descrevemos características necessárias em um sistema para seleção de pacientes por critérios clínicos utilizando dados de prontuário eletrônico. Implementamos um sistema de integração de dados baseado em reescrita de consultas, o que garante a obtenção de resultados sempre atualizados. O sistema utiliza ontologias para representação de conhecimento e é capaz de realização de inferência sobre diversos bancos de dados relacionais, por meio de expansão de consultas. Aplicamos esta implementação a um caso de estudo real e verificamos que a performance do nosso sistema é compatível com outros sistemas de integração e facilitaria o processo de seleção de pacientes para futuros estudos.

2.2 ORGANIZAÇÃO DO TEXTO

Este trabalho está dividido como se segue: no capítulo Revisão Bibliográfica e Fundamentos Conceituais, revisamos trabalhos que tratam do recrutamento de pacientes para estudos clínicos e abordamos tópicos relevantes de Ciência da Computação (alguns são discutidos em maiores detalhes nos apêndices). A seguir, em Especificação e Implementação, descrevemos o problema de maneira generalizada, apontamos soluções computacionais adequadas e descrevemos como implementamos nossa solução, o *Ontocloud*.

Em Avaliação, mostramos a aplicação de nosso método em uma situação prática, simulando o recrutamento de pacientes para um estudo clínico verdadeiro utilizando dados reais do AC Camargo Cancer Center. Comparamos ainda os resultados encontrados com resultados de outras soluções de integração de dados para avaliar seus diferenciais.

3 REVISÃO BIBLIOGRÁFICA E FUNDAMENTOS CONCEITUAIS

3.1 RECRUTAMENTO EM ESTUDOS CLÍNICOS

O recrutamento do número necessário de participantes de pesquisa é um desafio enfrentado por pesquisadores no mundo inteiro - estima-se que apenas 50% dos estudos clínicos atingem sua meta de recrutamento, e destes, apenas 50% o faz dentro do prazo estabelecido. Os problemas encontrados para o recrutamento variam desde aqueles relacionados ao paciente, ao seu médico, o centro de pesquisas, a organização e mesmo o desenho do estudo (Fletcher et al. 2012).

O trabalho Prokosch e Ganslandt (2009) apresenta uma visão geral de projetos que visam a criação de repositórios de dados (*data warehouse*) para consulta utilizando dados de prontuário eletrônico (coletados rotineiramente) com a finalidade de fomentar a pesquisa científica e estudos clínicos. São cinco os principais desafios identificados: (1) a construção do repositório de dados clínicos, (2) a elaboração de ferramentas para consultas *ad-hoc* [1], (3) estabelecimento de bancos de dados para pesquisa clínica, (4) ferramentas para apoio ao recrutamento dos pacientes e (5) utilização de dados de prontuário eletrônico na construção de bancos de dados para pesquisa. Destes pontos, alguns merecem destaque especial.

Publicações sobre *data warehouses* de dados clínicos mostram trabalhos com escopo limitado, por exemplo no trabalho Rubin e Desser (2008),

[1] Consultas potencialmente únicas elaboradas pelo pesquisador, sem interferência de um especialista em bancos de dados.

apenas duas fontes de dados são utilizadas. Outros trabalhos descrevem esforços semelhantes nos domínios de gerenciamento de enfermagem (Junttila et al. 2007), apoio a programas de gerenciamento de doenças (Ramick 2001), análise de custo baseado em diagnóstico e tratamento (Muranaga et al. 2007) e avaliação de efeitos adversos (Zhang et al. 2007).

Os sistemas de informação da maioria dos hospitais, apesar de conter módulos que auxiliam em diferentes graus a pesquisa clínica, não tem integração com os dados de prontuário eletrônico nem resultados de exames de biologia molecular; apesar de alguns trabalhos (Sahoo e Bhatt 2003; Velázquez et al. 2004) pregarem significante melhoria de qualidade e redução de custos na captura eletrônica de dados em estudos clínicos, outros (Schmier et al. 2005; Welker 2007) relevam o alto risco e custo da transição entre papel e eletrônico, que já é considerável em hospitais e mais importante nos estudos clínicos, que tem complexidades próprias; e que as questões regulatórias envolvidas em estudos clínicos aumentam a complexidade das ferramentas e da prática se se pretende capturar informações no prontuário eletrônico, durante atendimentos de rotina.

O trabalho Köpcke et al. (2013b) analisou a disponibilidade no prontuário eletrônico de dados relacionados a critérios clínicos. Foram estudados critérios de 15 estudos clínicos desenvolvidos em 5 hospitais na Alemanha. Apenas estudos desenvolvidos no mesmo departamento que faz o diagnóstico dos pacientes, e que não sejam patrocinados pela indústria farmacêutica foram considerados. 55% dos critérios estavam disponíveis no prontuário eletrônico como campos, e 64% dos pacientes possuíam estes campos preenchidos, o que dá uma cobertura média de 35% de dados de pacientes existentes para critérios clínicos destes estudos. Este valor está em concordância com outros estudos semelhantes. Quanto aos tipos de critério, ao considerar aqueles que especificam idade e gênero, 90% dos pacientes possuem esta informação; dados sobre a doença em tratamento atual existia para 60%; e dados

sobre comorbidades e medicações em uso estiveram presentes para apenas 10% dos pacientes. Os dados restantes foram encontrados como texto livre, no próprio prontuário eletrônico ou em papel. O mesmo autor, em outro trabalho (Köpcke et al. 2013a), construiu uma consulta no banco de dados de busca (*data warehouse*) do Erlangen University Hospital para identificar pacientes elegíveis para o estudo. Este banco de dados de busca foi construído em 2005, com solução proprietária e baseada em bancos de dados relacionais, e compreende informações clínicas e operacionais (como faturamento e procedimentos). Este estudo requeria que 510 pacientes fossem avaliados em uma semana, e que estes tivessem formulários preenchidos, com um mínimo de 46 e um máximo de 186 campos. Os resultados da consulta foram comparados com aqueles obtidos pela avaliação manual dos pacientes, e alguns dos formulários tiveram os campos previamente preenchidos com informações do banco de dados de rotina. A consulta ao banco de dados permitiu a inclusão de outros 42 (14%) de pacientes que não foram identificados manualmente, e também verificou que 21 (7%) foram incorretamente incluídos. A mediana do tempo de preenchimento dos formulários em branco foi de 255s, e aqueles que já vinham com alguns dados preenchidos do banco de dados tiveram mediana de 30s. O estudo conclui que o uso de dados de rotina para identificação de pacientes elegíveis e no preenchimento dos formuários de pesquisa economizou tempo e melhorou a inclusão de pacientes, porém não avalia os aspectos regulatórios do reuso de informações nem a qualidade destas.

No trabalho Dugas et al. (2010), desenvolvido no Universitätsklinikum Münster na Alemanha, utilizou-se uma ferramenta de relatórios disponível no sistema de prontuário eletrônico para buscar por pacientes internados de acordo com diagnóstico CID-10, departamento responsável pelo paciente, idade, e diagnóstico descritivo. Os critérios de inclusão e exclusão para sete estudos foram transformados em consultas (em linguagem própria do sistema) e diariamente eram executadas, procurando por pacientes internados no dia anterior

que atendessem a algum dos critérios especificados, e enviando e-mail aos pesquisadores responsáveis sempre que um paciente prospectivo era encontrado. O estudo ressalta que informações importantes para o recrutamento de pacientes em estudos de câncer, como tipo histológico e marcadores moleculares, não puderam ser incluídos nas consultas por estarem em outro sistema. Isso também indica que não foi feito nenhum esforço no sentido de utilizar informações de mais de um sistema utilizado naquele hospital. Também, relatou-se que o sistema não pode avaliar pacientes de ambulatório, uma vez que estes não recebem diagnóstico codificado. Dos 7 estudos, variou de 12% a 85% de aumento no número de pacientes recrutados.

Em Kamal et al. (2005), realizado no Ohio State Medical Center, foi utilizado um *data warehouse* baseado em tecnologia relacional já existente na instituição, que reunia informações de diversos softwares diferentes, para preparar uma ferramenta para pesquisadores buscarem pacientes prospectivos para estudos em câncer de mama. O estudo Weiner et al. (2003) implementou um sistema no Children's Hospital Boston que notifica automaticamente o pesquisador responsável quando um paciente que pode ser incluído em um estudo clínico é identificado pelas informações constantes do banco de dados. Foi elaborado um conjunto de critérios de pré-seleção, composto pelo diagnóstico, temperatura inferior a 38.5^{o}C, idade entre 10 e 21 anos e dor devido a vaso oclusão. A equipe médica e de enfermagem da emergência é instruída a informar o pesquisador responsável pelo estudo pelo *pager* quando um paciente que se enquadre nesses critérios aparece. O sistema criado acompanha cada paciente que dá entrada na emergência por oito horas em busca das informações necessárias para atender a este critérios, e em caso afirmativo, envia uma mensagem automática para o *pager* do pesquisador responsável. O sistema foi implantado 11 meses após o início do estudo e operou por 10 meses consecutivos, e verificou-se que a taxa de notificação (número de pacientes notificados dividido pelo número de pacientes elegíveis) antes de sua

introdução era de 56% (34 pacientes de 61), e após a entrada do sistema em produção, saltou para 84% (41 pacientes de 49).

O trabalho Fink et al. (2004) descreve um sistema baseado em ontologias que depende da entrada manual dos dados de um paciente. O objetivo do sistema é avaliar, dentre um grande número de estudos clínicos, quais deles o pacientes poderia ser incluído. Uma das conclusões deste estudo é que o recrutamento de um projeto de pesquisa pode ser aumentado em até 250% quando há o uso de ferramentas automatizadas para triagem de pacientes.

Um aspecto importante dos trabalhos acima é a necessidade implícita de se considerar dados provenientes de diversos sistemas de informação diferentes. Os trabalhos Köpcke et al. (2013a) e Kamal et al. (2005) basearam-se em *data warehouses* já existentes em suas instituições; Dugas et al. (2010) utilizou dados de apenas um sistema, e relatou a necessidade de integrar outros para atender a determinados estudos em oncologia; Weiner et al. (2003) baseou-se em dados de apenas um sistema, mas limitou-se a recrutar pacientes para um estudo sobre pacientes de emergência; e Fink et al. (2004) exige que os dados sobre o paciente sejam preenchidos manualmente. Além disso, o trabalho Beale (2005) mostra que, devido à especialização dos domínios em Medicina, é comum o uso de sistemas de informação específicos para cada especialidade, cada um com sua tecnologia e modelagem de dados diferente.

3.2 REPRESENTAÇÃO DE DADOS E CONHECIMENTO

3.2.1 Bancos Relacionais

O modelo relacional de dados, descrito pela primeira vez em Codd (1970), aborda o problema de tornar programas de computadores menos dependentes da forma como seus dados são armazenados, garantir um nível mínimo de consistência entre as informações, e acelerar as operações de armazenamento e procura por meio de indexação.

Ao criar um banco de dados, as tabelas e colunas representarão conceitos envolvidos no problema que se deseja resolver. Uma forma de organizar esses conceitos é criando um *modelo conceitual*. Uma das técnicas é a construção de um diagrama entidade-relacionamento, que agrega informações semânticas ao modelo relacional (Chen 1976).

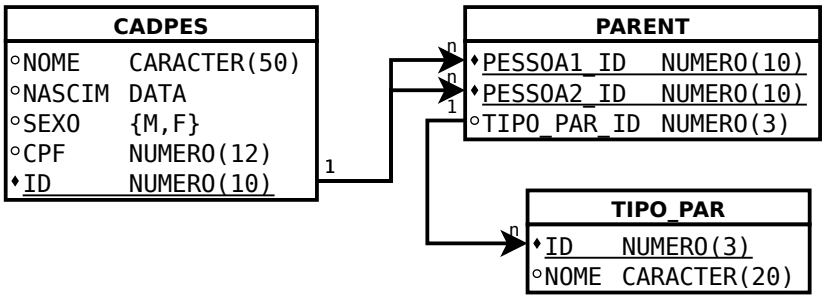

Figura 1 – Exemplo de tabelas em um banco de dados relacional.

Nesse modelo, os dados são organizados em tabelas, cada qual contendo colunas rotuladas e com um domínio correspondente. Os dados armazenados nessas estruturas são organizados em linhas. Na nomenclatura da teoria de bancos de dados relacionais, esses termos são equivalentes a relação, tuplas e atributos. Um exemplo de estrutura de banco pode ser visto na Figura 1, com dados no Quadro 1.

Quadro 1 – Exemplo de dados em três tabelas de um banco relacional.

CADPES

NOME	NASCIM	SEXO	CPF	ID
John Lennon	09-10-1940	M	-	1
Yoko Ono	18-02-1938	F	-	2
Sean Lennon	09-10-1975	M	-	3

TIPO_PAR

ID	NOME
1	CASADO
2	PAI
3	MÃE

PARENTES

PESSOA1_ID	PESSOA2_ID	TIPO_PAR_ID
1	2	1
1	3	2
2	3	3

Nesse exemplo, temos a tabela CADPES, que significa "Cadastro de Pessoas". Nela, a coluna NOME representa o nome de uma pessoa, e tem como domínio o tipo CARACTER(50), que representa uma sequência de até

50 caracteres. A coluna NASCIM representa a data de nascimento desta pessoa, e os campos SEXO e CPF podem ser interpretados de maneira semelhante. O campo ID (marcado com um losango cheio à esquerda, ao contrário dos outros campos) é uma *chave primária*, significando que identifica de maneira unívoca cada linha desta tabela. A tabela PARENTES ("Parentescos") relaciona duas pessoas entre si, por meio das colunas PESSOA1_ID e PESSOA2_ID. O valor desses campos deve ser relacionado com a linha que tenha o mesmo valor de ID, como indica as setas relacionando esses campos. O campo TIP_PAR_ID, por sua vez, vincula a relação de parentesco entre as duas pessoas com TIP_PAR, que lista os tipos de parentesco registrados no sistema.

Uma importante característica do modelo relacional é permitir que os programas acessem os dados utilizando os nomes das relações e colunas, ao invés de seus números de índice (como era o padrão na época em que foi proposto). Assim, mudanças na estruturação do banco de dados, desde que preservem os nomes, podem passar desapercebidas para os programas e usuários que os acessam. Além disso, programas diferentes podem acessar o mesmo banco de dados, desde que conheçam os identificadores de tabela e colunas.

O tempo gasto na inserção de novas linhas a uma tabela usualmente é constante, independente do número de linhas já existentes. As consultas, ou buscas por linhas que tenham certos valores (como por exemplo, “JOHN LENNON” na coluna NOME), tornam-se cada vez mais lentas, pois o algoritmo básico consiste em varrer todas as linhas em busca desse valor. Para mitigar esse problema são criados *índices*, estruturas que contém uma cópia dos valores da coluna indexada organizados de forma a acelerar a busca. Porém, uma vez indexada uma coluna, a inserção de novos valores torna-se mais lenta, uma vez que os índices vinculados àquela tabela devem ser atualizados.

Consultas são operações de busca dentro de uma ou mais relações,

em que se especificam condições para valores de domínios. Uma consulta em nosso modelo de exemplo pode ser encontrar todas as linhas da tabela PARENTES na qual uma das pessoas relacionadas é a de ID 1. Em linguagem de consulta SQL[2], a busca especificada poderia ser escrita como:

```
SELECT PESSOA1_ID, PESSOA2_ID, TIPO_PAR_ID
FROM  PARENTES WHERE PESSOA1_ID=1
```

Na consulta acima, SELECT indica que deseja-se recuperar uma ou mais linhas da tabela PARENTES (indicado pela cláusula FROM). Os campos da tabela que serão retornados são especificados logo após a cláusula SELECT. A palavra-chave WHERE indica uma condição lógica para que os registros sejam exibidos, no caso, um valor específico do campo PESSOA1_ID.

Grande parte dos sistemas de informação disponíveis atualmente, sejam proprietários ou livres, utilizam-se de bancos de dados relacionais para armazenamento(Hughes et al. 1999). Trata-se de uma tecnologia estável e bem estabelecida, e cujo mercado movimentou mais de 24 bilhões de dólares mundialmente em 2011[3] [4].

Até agora, falamos da teoria de representação de dados segundo o modelo relacional. Um sistema gerenciador de banco de dados relacional (SGBDR) é um software que implementa o modelo relacional (total ou parcialmente) e permite adicionar, remover e consultar tabelas. Grande parte dos softwares profissionais em operação hoje em dia faz uso de SGBDRs que utilizam a linguagem de consulta SQL (ou um de seus dialetos). As linguagens de consulta SQL são teoricamente embasadas pelas álgebras relacionais e cálculo relacional, que definem formalmente as operações de consulta. Inicialmente foi proposta como linguagem de consulta ao modelo relacional, e são

[2] http://www.iso.org/iso/catalogue_detail.htm?csnumber=45498

[3] http://www.gartner.com/id=1797615

[4] http://itknowledgeexchange.techtarget.com/eye-on-oracle/oracle-the-clear-leader-in-24-billion-rdbms-market/

um subconjunto da lógica de primeira ordem, especificamente são cláusulas de Horn sem negação ou recursão.

Apesar de representar um grande avanço em relação ao *status quo* da época e de ser atualmente a tecnologia dominante em representação de dados, as premissas do modelo relacional apresentam limitações que dificultam sua implantação em alguns ambientes. Na prática, é difícil modelar os dados de aplicações reais utilizando o modelo relacional, especialmente quando é necessário representar estruturas hierárquicas, lidar com dados faltantes e outras situações. Mudanças estruturais no modelo de dados geralmente refletem em alterações severas nos softwares que utilizam esse banco. O paradigma original evita que estas alterações sejam necessárias no caso de alterações em tipos de dados ou indexação, porém simplesmente trocar uma coluna de tabela ou mudar seu tipo de dado pode requerer uma refatoração completa do software.

Num banco relacional, as tabelas contém um número fixo de colunas; isto permite implementações mais eficientes se as consultas utilizarem sempre as mesmas colunas. Caso haja a necessidade de registrarem-se informações que não se aplicam para todas as colunas, é necessário o uso de colunas que ficam vazias em alguns casos ou novas tabelas, cada vez mais reduzindo a performance do banco. Em geral, num banco relacional, o dado é representado de forma muito próxima à representação interna de um programa de computador, o que em geral leva a soluções mais velozes porém menos flexíveis.

As estruturas de dados utilizadas pelas linguagens de programação não têm correspondência direta com o modelo relacional. Em particular, desde a década de 1990 é preponderante o uso de linguagens orientadas a objeto, que estabelecem um modelo de dados mais flexível e detalhado que o relacional. O modelo orientado a objeto especifica encapsulamento, herança,

diferentes níveis de acesso e polimorfismo, para proporcionar menor acoplamento e maior coesão ao programa, o que facilita sua extensão e manutenção. A diferença de representação entre o modelo orientado a objetos e o modelo relacional é conhecida como impedância relacional-objeto.

Além disso, o modelo relacional trabalha com dados; isso requer do usuário que fará a consulta conhecer o significado (semântica) das tabelas e colunas, além das características computacionais do domínio. O modelo relacional não prevê explicações descritivas sobre o significado de cada dado (anotação). Assim, elaborar consultas a um modelo relacional requer não só conhecimento técnico mas também acesso a documentação específica do sistema.

3.2.2 Ontologias

O termo “ontologia” tem sua origem no grego antigo, sendo “On” o prefixo para “entidade”. Ontologia, portanto, é o estudo das entidades reais ou abstratas e seus relacionamentos.

Podemos definir uma ontologia como uma estrutura que representa conhecimento de uma maneira formal, como conceitos e relacionamentos entre os mesmos (Brachman e Levesque 2004), ou uma especificação explícita de uma conceitualização (Gruber 1993). Uma conceitualização é uma visão abstrata e simplificada do mundo que se deseja representar para algum propósito. Todo campo de estudo ou sistema de informação está vinculado a um padrão de conceitualização, seja de forma implícita ou explícita. Em sistemas de informação baseados em conhecimento, o conceito de “existência” é definido como “aquilo que pode ser representado”.

Esses objetos, e as relações entre eles que podem ser descritas, refletem-se no vocabulário representacional em que esse sistema de informação representa conhecimento. Nessa ontologia haverá definições que relacionam elementos do universo de discurso com texto descritivo (que tem o objetivo de

descrever o significado desses elementos), e axiomas formais que restringem o uso desses objetos (Gruber 1993).

O exemplo de banco relacional do Quadro 1 pode ser expresso como a ontologia expressa no Quadro 2.

Quadro 2 – Versão do exemplo de cadastro de pessoas representado como ontologia.

$Homem(johnLennon)$
$Homem(seanLennon)$
$Mulher(yokoOno)$
$dataDeNascimento(johnLennon, `09-10-1940')$
$dataDeNascimento(seanLennon, `09-10-1975')$
$dataDeNascimento(yokoOno, `18-02-1938')$
$nome(johnLennon, `JohnLennon')$
$nome(seanLennon, `SeanLennon')$
$nome(yokoOno, `YokoOno')$
$casadoCom(johnLennon, yokoOno)$
$pai(seanLennon, johnLennon)$
$mae(seanLennon, yokoOno)$

As ontologias distinguem suas entidades de acordo com diversos critérios:

- Classes e instâncias: as instâncias são objetos, ou indivíduos, que podem pertencer a uma ou mais classes (ou categorias). Por exemplo, a instância $johnLennon$ pertence à classe $Pessoa$.
- Propriedades: são entidades que relacionam instâncias ou classes entre si. A propriedade mae relaciona duas instâncias de $Pessoa$, de forma que a primeira é filha da segunda.
- Axiomas: relacionam as entidades em fórmulas lógicas, de maneira formal. Por exemplo, $Homem$ é subclasse de $Pessoa$ ($Homem \sqsubseteq Pessoa$).

Da mesma forma, classes podem conter subclasses, significando que todos os elementos desta última pertencem também à primeira, compartilhando suas propriedades. A classe $Pessoa$ contém a subclasse $Musico$, que contém a subclasse $Guitarrista$.

Instâncias podem pertencer a diversas classes, porém podem existir regras restringindo estas relações. Por exemplo, se uma instância que pertença à classe $Homem$ não pode pertencer ao mesmo tempo à classe $Mulher$, dizemos então que estas classes são disjuntas. Existe também a partição, em que membros de uma categoria necessariamente são membros de uma e apenas uma subcategoria (como por exemplo, $Pessoa$ e as subcategorias $Homem$, $Mulher$ ou $Intersexo$ (XXY,XYY)). Decomposição exaustiva exige que membros de uma categoria sejam membros de uma ou mais subcategorias.

Além disso, as ontologias podem representar medidas físicas (peso, altura, velocidade) em unidades diferentes, objetos contáveis e incontáveis (como moedas e líquidos), processos e passagem do tempo. Mais complexo é representar diferentes crenças que determinados agentes (como pessoas ou softwares) possuem sobre a mesma realidade, ou estados não determinísticos (e conhecimento parcial sobre a realidade) (Russell e Norvig 1995).

Inferência é o processo pelo qual nova informação é derivada de declarações e axiomas (Russell e Norvig 2003). No exemplo dado, $johnLennon$ e $seanLennon$ são instâncias da classe $Homem$. Esses fatos, juntamente com o axioma que declara $Homem$ subclasse de $Pessoa$, permite inferir que estas instâncias são também instâncias da classe $Pessoa$.

3.2.3 Comparação das metodologias de representação de informação

Bancos de dados relacionais representam dados, enquanto que as ontologias expressam conceitos, e isso pode levar usuários do primeiro a cometer enganos na intepretação com mais frequência que dos últimos. No exemplo do Quadro 1, a tabela PARENTES relaciona duas linhas da tabela CADPES com uma linha da tabela TIP_PAR, porém nada impede que se crie uma relação direta entre a tabela CADPES e a TIP_PAR, utilizando-se do campo CPF da primeira e do identificador do segundo. Já em uma ontologia, o relacionamento entre entidades é feito por meio de URIs, e não apenas

códigos numéricos, o que impede que relações sem nexo sejam estabelecidas. No exemplo abaixo, expressamos em RDF algumas das informações do Quadro 1.

```
PREFIX : <http://exemplo.com/>
:individuo1 :nome "John Lennon";
            :dataDeNascimento "09-10-1940";
            :casadoCom :individuo2.

:individuo2 :nome "Yoko Ono".
```

Nessa representação, a propriedade $casadoCom$ relaciona dois recursos, $individuo1$ e $individuo2$. Aliado ao prefixo estabelecido, não é possível intepretar a relação entre os indivíduos de outra forma. Na contrapartida relacional, o relacionamento é feito por meio de identificadores puramente numéricos, que poderiam indicar linhas em qualquer tabela. Esta diferença entre a semântica de bancos de dados relacionais e ontologias é fundamental para entender características chave destas últimas que serão úteis para integração de dados.

Uma diferença importante ao comparar as tecnologias é a definição de *mundo aberto* e *mundo fechado*. Em sistemas baseados em conhecimento que assumem a premissa de mundo fechado, como os bancos relacionais, se uma informação não está contida na base de dados, então ela é falsa. Por exemplo, no exemplo anterior, se questionarmos se johnLennon é um baixista, a resposta será “não” sob a premissa de mundo fechado, pois esta informação não consta na base de dados. Porém, na premissa de mundo aberto, seguida pelos sistemas baseados em ontologias, a resposta será “desconhecido”, pois em outra base de dados esta informação pode existir. Nessa premissa, apenas

se for afirmado que johnLennon não é um baixista ($\neg Baixista(johnLennon)$), a resposta seria "não"[5].

Formatos de representação de conhecimento, como OWL, especificam semântica em suas propriedades e classes, o que permite que afirmações sobre classes sejam propagadas como afirmações sobre instâncias particulares. Em bancos de dados relacionais, o recurso de *colunas virtuais*, implementado em alguns sistemas, permite que algumas colunas tenham um valor dado por uma expressão calculada dinamicamente; visões também podem calcular expressões matemáticas e emular inferência. Por mais que desta maneira se possa chegar a resultados equivalentes aos obtidos por meio de inferências em uma ontologia, a maneira como o conhecimento é descrito é demasiadamente vinculado ao modelo de dados utilizado, o que limita sua utilização em outras situações.

Quanto às linguagens de consulta, SPARQL e SQL diferem não só no modelo de dados, mas em especial a cláusula SERVICE confere ao primeiro a capacidade de integrar resultados de diferentes documentos RDF. Não há no padrão SQL especificação semelhante. Os sistemas de bancos de dados relacionais que implementam federação de bancos de dados criam extensões do padrão, que são próprias de cada sistema.

3.2.4 Acesso a dados baseado em ontologias (OBDA)

Quando há necessidade de acessar bancos de dados relacionais como RDF ou ontologias, aplica-se uma camada de software conhecida como Acesso a dados baseado em ontologias (OBDA). Apesar de ser possível representar os mesmos dados em bancos relacionais ou ontologias, é necessário criar um mapeamento entre os conceitos de ambos, adicionando semântica ao banco relacional. Esse mapeamento é utilizado então para converter a totalidade dos

[5] É indiferente se esta afirmação é feita diretamente ou indiretamente, como no axioma $Baixista \sqcap Guitarrista \sqsubseteq \bot$, ou "não existe ninguém que seja baixista e guitarrista ao mesmo tempo".

dados do banco relacional em uma ontologia (método baseado em replicação de dados), ou traduzir consultas feitas à ontologia em consultas ao banco relacional (método baseado em reescrita de consultas). Este último método traz resultados sempre atualizados, refletindo os dados mais recentemente cadastrados no banco relacional, enquanto que o baseado em replicação de dados traz o estado do banco em algum ponto fixo passado e deve ser constantemente atualizado. Entretanto, as consultas apresentadas ao sistema baseado em reescrita de consultas devem passar por etapas de tradução e são necessariamente mais lentas que as apresentadas a sua contrapartida (Figura 2).

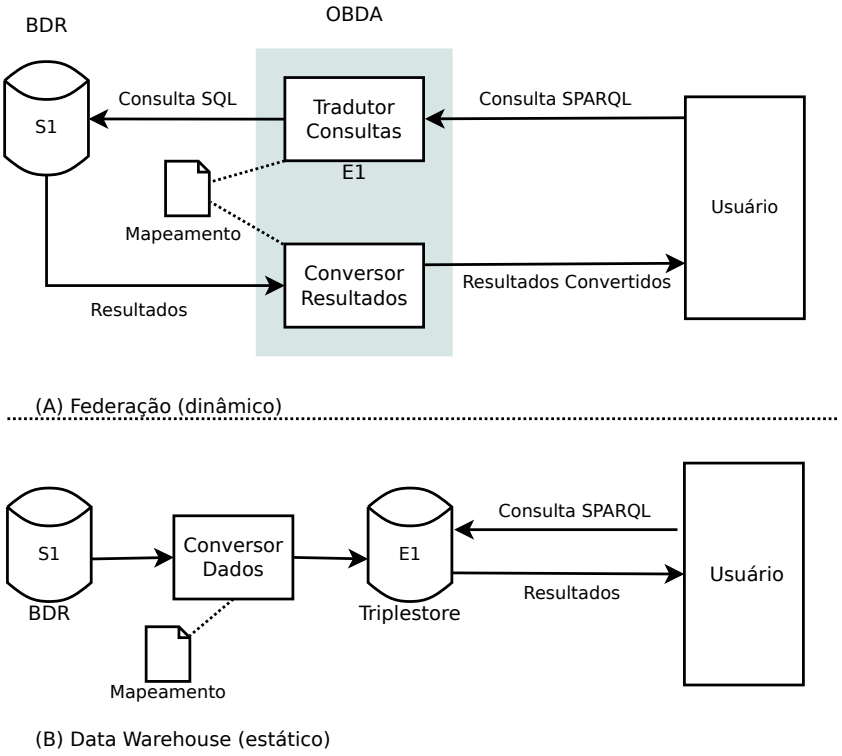

Figura 2 – Comparação de métodos de integração baseados em reescrita de consultas e replicação de dados.

3.3 INTEGRAÇÃO DE DADOS HETEROGÊNEOS

A integração de dados heterogêneos tem como objetivo responder consultas que utilizem informações de diversas origens diferentes. As principais

características das fontes de dados que devem ser analisadas ao elaborar um sistema de integração são:

- Dinâmica das fontes: qual a frequência de inclusão/remoção de fontes do ambiente de integração?
- Frequência de atualização das fontes: os dados contidos em cada fonte são atualizados com que frequência?
- Integração ponto a ponto ou centralizada: as fontes de dados precisam trocar informações entre si? Ou é necessário um repositório central de informações consolidadas?
- Requisitos não funcionais: as consultas ao sistema de integração precisam trazer resultados refletindo o estado atual dos dados? Qual o intervalo de tempo aceitável entre atualização na fonte e resposta?

Caracterizar o problema de integração quanto a estas variáveis é fundamental para a criação de um sistema de integração adequado.

Num ambiente em que as fontes de dados mudam dinamicamente, como uma rede de celulares, uma estrutura extremamente flexível deve ser disponibilizada para manter atualizada a lista de fontes atualmente registrada. Já num sistema de integração de informações corporativas, esta flexibilidade não é necessária, uma vez que novos sistemas são introduzidos com frequencia muito reduzida.

Em determinados problemas, é tolerável a obtenção de resultados desatualizados ao consultar um sistema de integração. Nessas ocasiões, pode-se considerar a criação de um repositório contendo uma cópia traduzida dos dados existentes nas fontes. Esse método é conhecido como *data warehousing* (ou *Extract-Transformation-Load*, ETL). Caso contrário, as consultas apresentadas ao sistema de integração devem ser traduzidas em consultas às fontes de dados, o que traz resultados atualizados porém implica em custo

computacional maior (o que pode prejudicar a operação dos sistemas). Esse processo é conhecido como *integração de visões* (Casanova e Vidal 1983).

Integração ponto a ponto supõe que as fontes de dados terão seus dados traduzidos e intepretados entre si, e portanto um formato comum de comunicação ou mapeamentos entre as fontes são necessários. No caso da criação de um esquema global (um repositório central com a informação consolidada em um formato comum), o objetivo é permitir consultas expressas em um formato comum e que retornem resultados de todas as fontes de dados devidamente traduzidos.

Com relação a entidades descritas pelas fontes de dados, a integração pode ser feita verticalmente, quando há dados sobre uma mesma entidade espalhada em diversas fontes, ou horizontamente, quando diferentes entidades são descritas por cada uma das fontes de dados.

Para exemplificar, considere que um hospital possua dois bancos de dados clínicos, um contendo dados obtidos nos atendimentos médicos, e outro com resultados de exames de laboratórios. Ao buscar por pacientes que tenham passado por consulta numa dada especialidade e que tenham feito exames de raio X, deve-se fazer uma integração horizontal, pois os dados de atendimento e de exame referem-se à mesma entidade, o paciente. Em outra situação, em que deseja-se saber quais pacientes realizaram um dado exame em diversos laboratórios, a integração é vertical, uma vez que cada fonte de dados contribui com diferentes entidades e com a mesma informação para cada um deles.

Na maioria das situações práticas, integração vertical e horizontal ocorrem simultaneamente; no primeiro exemplo dado, podem existir pacientes que não passaram por atendimento mas realizaram exame, e no segundo, pacientes que realizaram exame em dois laboratórios diferentes.

Além disso, as fontes de dados podem conter, em linhas gerais, as mesmas informações, porém em formatos diferentes e requerer um esforço

de consolidação para serem efetivamente integrados. Estas diferenças são chamadas de heterogeneidades de dados e são classificadas e resolvidas de diferentes maneiras.

3.3.1 Níveis de heterogeneidade

A maior barreira à integração de bancos de dados é, sem dúvida, a multiplicidade de formas nas quais informações similares podem ser representadas em um computador. As diferenças podem ser sintáticas, esquemáticas e semânticas, cada qual se referindo a um aspecto da representação de informação (Beck et al. 2008).

O diagnóstico do paciente, por exemplo, pode ser encontrado na maioria dos sistemas informatizados na forma da codificação CID-10, pois esse código é usado para propósitos de faturamento junto às operadoras de saúde e portanto está presente nos sistemas informatizados. Porém, os critérios de inclusão podem especificar subclassificações da doença que não são codificadas de maneira unívoca pelo CID-10. Outras informações estão presentes no PEP em meio a textos discursivos, sujeito a variações de grafia e erros de digitação, o que exige soluções de um campo específico de pesquisa em ciência da computação conhecido como Processamento de Linguagem Natural (NLP).

3.3.1.1 Heterogeneidade sintática

Heterogeneidade sintática é uma categoria ampla de diferenças na formatação de dados semelhantes. Por exemplo, documentos em formato XLS e SPSS podem conter dados idênticos, mas cada um é descrito por sintaxes diferentes.

Há muitos níveis sintáticos envolvidos no armazenamento e representação de dados em um sistema de informação. Hoje em dia, os sistemas de

informação são baseados em implementações padronizadas, minimizando algumas questões de heterogeneidade sintática. Podemos citar como exemplo a padronização existente em transmissão de arquivos, dados e de transmissão/recepção.

O primeiro requisito para a integração é homogeneizar o acesso às informações em nível sintático, que será definido pelos projetistas do sistema de integração; no caso de um sistema de integração baseado em dados estruturados, sejam bancos relacionais, orientados a objetos ou híbridos, isto significa prover acesso unificado às diferentes fontes sem modificações em suas estruturas.

3.3.1.2 Heterogeneidade esquemática

Heterogeneidade esquemática refere-se às diferenças na estrutura em que os dados estão organizados. Por exemplo, um sistema de gestão hospitalar pode armazenar os dados de médicos e enfermeiras em uma só tabela, com uma coluna que especifique seu cargo, e outro sistema pode utilizar tabelas diferentes.

Diferenças nos tipos de dado também devem ser levadas em conta nessa etapa. Num caso, cada pessoa pode ser identificada por uma sequencia de letras e números (o médico identificador por uma letra “M” seguida de um número de 5 dígitos) e em outro, apenas um número. Bancos de dados podem ser modelados de muitas maneiras diferentes e ainda assim atender aos requisitos funcionais da aplicação a que se destina. Assim, aplicações diferentes com a mesma finalidade podem ter seus dados representados de maneira muito diferente e assim não serem interoperáveis.

3.3.1.3 Heterogeneidade semântica

Duas tabelas identicamente esquematizadas podem ainda assim apresentar diferenças que dificultam, ou impossibilitam, sua harmonização: a heterogeneidade semântica aparece quando há diferença na interpretação de um valor presente. Heterogeneidade semântica é um problema intrínseco a integração de bancos de dados e que geralmente causa perda de especificidade da informação (Hull 1997; Sujansky 2002).

Outro tipo de heterogeneidade semântica aparece quando os níveis de detalhe dos dados (granularidade) são diferentes. Há duas soluções possíveis para esse tipo de heterogeneidade semântica: utilizar, na visão reconciliada, o menor nível de detalhe possível, ou harmonizar os dados de diferentes detalhes em uma estrutura hierárquica, como demonstrado no exemplo da Figura 3. Em (A), vemos duas tabelas, *db1* e *db2*, ambos representando opções para uma pergunta sobre hábito de fumar. Porém, as opções de *db1* tem descrições subjetivas (“uso leve” ou “uso pesado”), enquanto que *db2* tem opções objetivas, descrevendo a quantidade de maços de cigarro fumados por dia. Assumindo-se que o uso de até um maço por dia é considerado um uso leve de tabaco, e uso pesado a partir de dois maços por dia, podemos fazer um mapeamento e traduzir os dados constantes em *db2* para *db1*. Esse mapeamento é possível, pois para cada opção existente em *db2*, corresponde uma única opção de *db1*, porém há perda de informação, já que as opções no último são mais detalhadas que no primeiro. O contrário não é possível, pois para cada opção em *db1* corresponde mais de uma opção em *db2*.

Reconciliação hierárquica é o processo em que conceitos com semânticas semelhantes mas não idênticas são relacionados formalmente. Na Figura 3 vemos um exemplo aplicado a perguntas e respostas. Na Figura 3B, os conceitos representados pelas opções na Figura 3A são organizados hierarquicamente, de maneira que os conceitos mais detalhados sejam vinculados

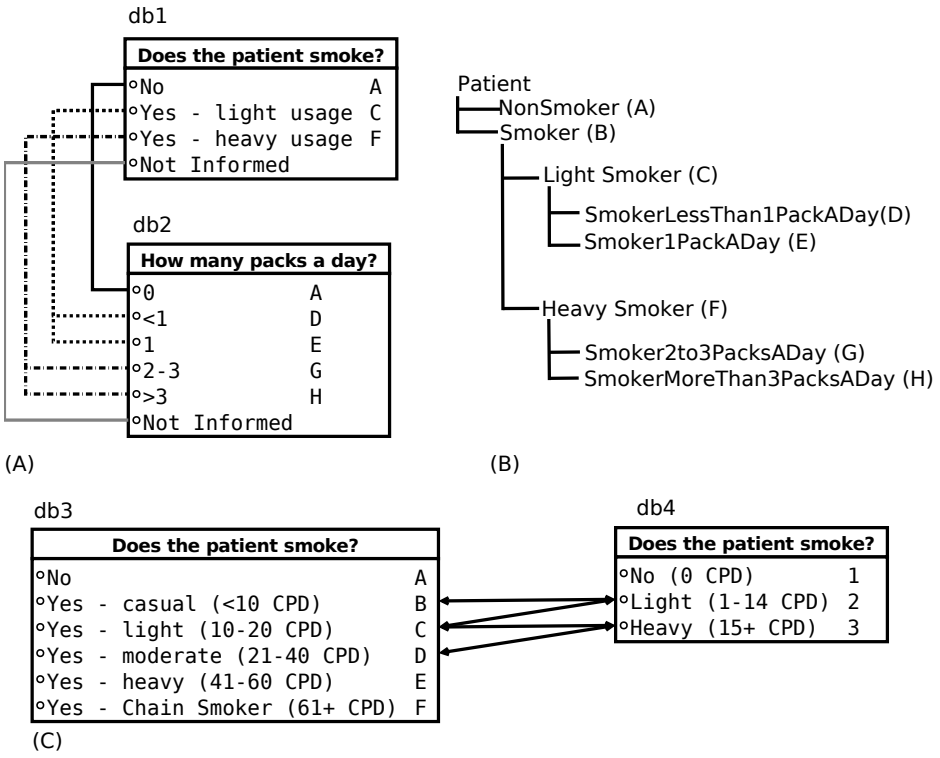

Figura 3 – Exemplo de um problema de reconciliação semântica.

aos mais genérico. Note que existe uma letra A-H para cada conceito tanto na Figura 3A e na 3B, e que esta letra identifica os conceitos semelhantes entre as duas representações. Assim, os dados de *db1* e os de *db2*, que tem diferentes graus de detalhamento, coexistem na mesma representação. A vantagem dessa organização é que, ao mesmo tempo que uma busca por conceitos mais gerais como “Smoker” (B) retorna os resultados de ambos bancos de dados, é possível também buscar pelos conceitos mais detalhados.

Existem casos em que não é possível harmonizar a representação dos dados: quando há intersecção conceitual entre os campos. Por exemplo, na Figura 3C, os bancos de dados *db3* e *db4* representam faixas de número de cigarros por dia (CPD) usados pelo paciente. A integração hierárquica é impossível, pois existem conceitos do banco *db3* que correspondem a mais de um conceito do banco *db4*, e vice versa. O conceito “Yes - light (10-20CPD)” de *db3* não pode ser mapeado com exatidão para “Light (1-14CPD)” nem “Heavy (15+CPD)” de *db4*, e o mesmo vale para estas duas últimas opções.

3.3.2 Replicação de dados/reescrita de consultas

Quanto à estratégia de acesso das informações num sistema de integração, tanto as fontes de dados podem ser traduzidas e consolidadas de antemão para um esquema harmonizado, que posteriormente será consultado, ou mediante uma consulta, as fontes de dados são consultadas e apenas os resultados traduzidos.

A primeira estratégia, conhecida como replicação de dados, *data warehousing* ou *Extract-Transform-Load* - ETL, tem a vantagem de prover um acesso mais veloz ao banco consolidado, e de ser mais simples de se construir. Porém, a visão consolidada representará uma cópia estática das fontes no momento em que foi construído; isso não representa um problema quando as fontes de dados não são atualizadas constantemente. Porém muitas vezes precisamos realizar a consulta ao banco consolidado baseado em informações recentes, e para cada consulta seria necessário realizar nova cópia do banco. Outra desvantagem pode ser o tempo empregado na geração da visão consolidada, e espaço extra consumido, se as tabelas fonte são muito grandes. Esta alternativa é conhecida como a criação de um *data warehouse* (Inmon 2005).

A segunda estratégia, que chamamos reescrita de consultas, envolve transformar uma consulta à visão consolidada em diversas consultas às tabelas fonte. Apesar de gerar resultados recentes sem a necessidade de um processo de cópia demorado, o problema de transformar as consultas pode ser complexo e demorado, ou de difícil manutenção. Abaixo estão as duas principais estratégias para a consulta à chamada federação de bancos de dados que assumem a transformação dos dados por visões consolidadas ou tabelas fontes.

- Global as view (GAV): Esta é a estratégia em que cada visão consolidada é representada em função das tabelas das fontes. Aqui, o trabalho de

fazer a integração é deixado para o desenvolvedor, resultando em consultas às fontes que são mais rápidas. Entretanto, como uma só visão consolidada será representada pela união, muitas vezes complexa, de uma ou mais tabelas fonte, torna-se difícil adicionar novas fontes ao esquema integrado. Também, mudanças estruturais nas fontes requerem um esforço grande do programador, que terá que alterar a estrutura de todas as visões que a consolidam. Existem sistemas (Calvanese et al. 2007) que, a partir de mapeamentos GAV limitados (isto é, que traduzem em SQL apenas uma fonte), combinam os mesmos em uma consulta maior para realizar a integração. Esta metodologia evita os problemas de manutenibilidade da abordagem tradicional, mas tende a gerar consultas pouco eficientes.

- Local as view (LAV): Nessa estratégia, as tabelas fonte é que são modeladas em função da visão consolidada, e as consultas à esta última são transformadas em consultas às fontes por um processo conhecido como "transformação de consultas baseado em visões" (*query rewriting*); ou os resultados da query ao modelo global são calculados a partir dos resultados parciais que são obtidos nas tabelas fonte no processo "resposta a consulta baseada em visões"(*query answering*). O processo de consulta às fontes é complexo (ou, dependendo do tipo de consulta que será feita, impossível) e lento, mas como as consultas usadas para a consolidação são feitas para cada tabela fonte (e, portanto, não referenciam duas fontes separadas), o processo de adição de novas fontes e manutenção das antigas é muito simplificado.

3.3.3 Integração em banco relacional/ontologia

Outra decisão a ser tomada é quanto ao tipo de representação que será utilizado para representar o esquema global: os bancos relacionais ou ontologias. No contexto de um problema de integração de dados, algumas

características destas soluções são de grande importância e as detalharemos abaixo.

Num banco relacional, as tabelas contém um número fixo de colunas; nas ontologias, cada conceito pode ter definições completamente diferentes dos outros, porém a performance é pior, com relação a bancos relacionais. Quando, ao integrar bancos de dados, existem muitos dados faltantes, ou em granularidades diferentes, deve-se pesar a necessidade de maior expressividade e de performance.

Tem sido proposto o uso de ontologias para reconciliação semântica em integração de dados heterogêneos (Cruz e Xiao 2005; Wache et al. 2001). O problema da heterogeneidade semântica na integração de dados é mais facilmente lidável numa ontologia, pois podemos representar conceitos de diferentes granularidades de forma unificada, ou mesmo relacionar conceitos por meio de axiomas ou regras. No exemplo da Figura 3, a resolução por ontologias da heterogeneidade permite que, por meio dos axiomas $Smoker1PackADay \sqsubseteq LightSmoker$, unifiquemos os conceitos em ambos bancos de dados sem perdas. Num banco relacional, seria necessário que o conceito mais específico ($Smoker1PackADay$) fosse transformado no conceito menos detalhado ($LightSmoker$), perdendo-se detalhamento da informação.

Por fim, as ontologias são amplamente utilizadas no contexto da web semântica, na qual há disponibilização de informações em formato de ontologia e com anotações suficientes para permitir a compreensão do significado das informações contidas. Por exemplo, a ontologia *foaf*[6] descreve os conceitos envolvidos na identificação de pessoas, e tem sido utilizada por diversas ontologias que necessitem desta identificação. Assim, uma ontologia que utilize o conceito foaf:name contém o endereço na internet para a definição do significado desta propriedade. Da mesma forma, um código CID-10, no meio

[6] `http://www.foaf-project.org/`

médico, é universalmente reconhecido e não deixa dúvidas quanto a sua interpretação. A interpretação dos dados de um banco relacional depende de como o mesmo foi construído, documentado ou até mesmo da aplicação a que ele se destina, em suma, não foi feito tendo em vista a compreensão por um agente humano.

3.4 FERRAMENTAS PARA INTEGRAÇÃO DE DADOS

3.4.1 Métodos de replicação de dados

Na construção de um sistema de integração de bancos de dados baseado em replicação de dados (ou *data warehouse*), podem ser empregadas diversas ferramentas. O software Pentaho Data Integration[7] é um software de código aberto para facilitar o processo de integração de diversos tipos de dados, estejam em bancos relacionais, serviços web, arquivos ou outros. O Informatica PowerCenter[8] é uma ferramenta proprietária robusta para integração em nível corporativo, com características de escalabilidade para possibilitar boa performance mediante alto volume de dados.

3.4.2 Métodos de reescrita de consultas baseados no modelo relacional

Os métodos de reescrita de consultas baseados em bancos de dados relacionais são estudados e desenvolvidos desde 1979 (Batini et al. 1986). É chamada uma federação de bancos de dados uma “coleção cooperativa de sistemas de bancos de dados autônomos e possivelmente heterogêneos” (Sheth e Larson 1990). A federação (ou banco de dados virtual - *Virtual Data Base* ou VDB) se apresenta como um banco de dados que, por meio de mediadores, traduz consultas ao esquema global em consultas aos bancos de origem. Uma consulta complexa, envolvendo objetos presentes em diferentes bancos de dados com representações heterogêneas, deve ser reestruturada levando

[7] http://community.pentaho.com/

[8] http://www.informatica.com/us/products/data-integration/enterprise/

em conta o custo computacional de acesso a cada uma das origens referenciadas de uma forma específica para aquela coleção de bancos de dados, de forma a proporcionar uma boa performance (Haas et al. 2002).

Apesar de sua introdução ter sido há mais de duas décadas, o campo de federações de bancos de dados continua a ser objeto de pesquisa tanto no setor acadêmico quanto no tecnológico. O trabalho Yuhanna e Gilpin (2012) compara soluções para virtualização de bancos de dados proprietárias e de código aberto de acordo com 53 critérios. A tese Pullokkaran (2013) analisa aspectos técnicos e viabilidade econômica da virtualização de bancos de dados em comparação com métodos tradicionais de data warehousing,

Existem diversas ferramentas modernas que implementam bancos de dados virtuais. Das soluções proprietárias, podemos destacar o Oracle Heterogeneous Data Services, o IBM Infosphere e o Openlink Virtuoso. Das soluções open source, temos o Teiid[9], Mule ESB[10] e MySQL Federation Plugin. As soluções mencionadas permitem o acesso a outros bancos de dados de diferentes tecnologias utilizando mediadores próprios. Desta forma a integração de bancos de dados é implementada por visões que consolidam as diferentes origens de dados, utilizando metodologia de mapeamento GAV.

3.4.3 Métodos de reescrita de consultas baseados em ontologias

Existem diversos softwares e frameworks de integração de dados baseados em ontologia descritos na literatura. Wache et al. (2001) e Buccella et al. (2005) comparam diversas ferramentas desse tipo quanto a suas arquiteturas e características, como estratégia de integração (repositório central, ou distribuído), o tipo de linguagem ontológica utilizada e metodologia de engenharia de ontologia utilizado.

O trabalho de Calvanese et al. (2007) descreve um sistema de integração de dados baseados em reescrita de consultas e em ontologias chamado

[9] http://www.jboss.org/teiid/
[10] http://www.mulesoft.com/mule-esb-features

Mastro-I. Foi cuidadosamente elaborado para que o processo de tradução de consultas à ontologia em queries SQL nos bancos fonte mantenha-se computável, limitando a expressividade das consultas apresentadas à ontologia. A integração é realizada em duas etapas: primeiro, é utilizado um produto proprietário (IBM Infosphere Federation Server) que provê acesso unificado a diversos bancos de dados de diferentes tecnologias (no caso de uso apresentado no trabalho, bancos relacionais e arquivos XML foram utilizados); em seguida, esse banco heterogêneo é mapeado em conceitos ontológicos, utilizando uma linguagem de mapeamento própria. Os mapeamentos se enquadram na categoria GAV, pois são compostos por consultas na linguagem do banco de dados de origem que deve incluir os componentes que serão utilizados nas triplas formato RDF.

DBOM (Cure e Bensaid 2008) é um sistema de integração que utiliza fragmentos decidíveis da linguagem OWL, ou seja, OWL-DL e OWL-DL lite, para mapear resultados de consultas feitas a um banco relacional para uma ontologia. Diferentes fontes relacionais podem ser utilizadas ao mesmo tempo, não havendo necessidade de outra camada de resolução de heterogeneidade. Esse sistema lida com diferenças de confiabilidade entre fontes de informação por meio de um parâmetro especial definido em configuração. Está implementado como um *plug-in* do software *Protégé*. Como caso de uso, os autores apresentam a integração de dois bancos de dados de drogas homeopáticas, exemplificando a consolidação de dois conceitos semelhantes e como o grau de confiabilidade é utilizado nessa harmonização.

FedX (Schwarte et al. 2011a) é um framework para execução de consultas SPARQL federadas, podendo atuar sobre endpoints, repositórios SAIL e outros. Sua criação é anterior à especificação SPARQL 1.1, porém as últimas versões estão adequadas a esse padrão. Por meio de consultas ASK, o sistema checa quais propriedades e classes são servidos por cada origem

de dados e constrói o planejamento federado. Em seguida, realiza otimizações e reordenação das consultas utilizando algoritmo específico (Schwarte et al. 2011b). A execução das consultas também é específica para origens de dados distribuídas, pois paraleliza a execução das cláusulas JOIN e UNION existentes nas mesmas.

No trabalho Min et al. (2009), dois bancos de dados com dados clínicos sobre pacientes diagnosticados com câncer de próstata são integrados utilizando-se uma ontologia. São duas fontes de dados utilizadas, a do Registro de Câncer, que contém informações demográficas, e a do Departamento de Radioterapia, com dados clínicos e de radioterapia. Ambos foram integrados utilizando-se uma única instância de software de mapeamento de dados relacionais em ontologias D2R. Exceto pelo estadiamento TNM, os dados integrados são complementares.

O DebugIT Clinical Data Repository (Teodoro et al. 2009) é um framework de integração de repositórios de dados de uma rede de hospitais do projeto DebugIT (Detecting and Eliminating Bacteria Using Information Technology). Cada um dos participantes da rede deverá transformar e publicar seus dados num esquema relacional, comum a todos, seguindo as ontologias NEWT para bactérias, WHO-ATC para drogas, SNOMED CT para outras informações como culturas, TIME-OWL para atributos temporais, e outras. Após esta transformação, os dados são consolidados ainda em formato relacional num repositório virtual por um servidor MySQL com o plugin de federação. Por fim, esse banco de dados é publicado como um endpoint SPARQL por um servidor D2R (Bizer e Seaborne 2004), de acordo com a ontologia DCO (Schober et al. 2010). Como toda a integração do sistema é desempenhada por métodos relacionais e apenas no final ocorre a publicação como RDF, esta é uma aplicação de método relacional de federação que não desfruta das vantagens da ontologia para reconciliação semântica. As ontologias referidas são utilizadas apenas para normalização do vocabulário, e não do esquema global.

3.4.4 Comparação das metodologias

Para avaliar as diferentes soluções para integração de banco de dados baseadas em reescrita de consultas, verificamos atributos que categorizamos como objetivos e funcionais. Os atributos objetivos são aqueles podem ser avaliados por um experimento, como velocidade e acurácia dos resultados obtidos. Quanto aos atributos funcionais, são características da configuração do sistema de integração que não influenciam necessariamente os atributos objetivos mas podem facilitar ou dificultar a preparação e manutenção do sistema. Consideramos desta categoria: independência entre as fontes de dados, separação entre conhecimento médico e conhecimento sobre os bancos de dados e realização de inferência.

O Quadro 3 compara algumas das principais soluções para integração de bancos de dados descritas atualmente em função de algumas características. *Acesso dinâmico* refere-se à habilidade do sistema em acessar informações diretamente dos bancos de origem, trazendo os resultados mais recentes possíveis. O *acoplamento do mapeamento* pode ser forte ou fraco, indicando o quanto o mapeamento global das fontes depende de cada uma delas. *Reconciliação semântica* pode ter valores "Funcional"ou "Ontologia", significando por que meio a conciliação de conceitos com intersecção semântica é feita. *Camada de integração* descreve em que tipo de tecnologia a integração é efetivada, se numa camada de bancos relacionais (BDR), em ontologia ou numa solução híbrida. *Inferência em dados totais* significa a capacidade de realizar inferência utilizando-se da totalidade de informações disponíveis na federação (e não apenas em cada uma das fontes de dados), e de que maneira essa inferência é realizada, por funções ou por ontologias. Finalmente, *Mapeamento relacional* é a capacidade de traduzir dados presentes em bancos de dados relacionais, e se essa tradução é feita diretamente na fonte de dados, ou no esquema global, após a integração ter sido feita em banco relacional.

Quadro 3 – Comparação das características desejáveis de sistemas de integração.

Sistema	1- Relacional	2- FedX	3- Mastro-I	4- DebugIT
Acesso dinâmico	Sim	Sim	Sim	Não
Acoplamento do mapeamento	Forte	N/A	Fraco	Forte
Reconciliação semântica	Funcional	Ontologia	Ontologia	Funcional
Camada de integração	BDR	Ontologia	Ontologia+BDR	BDR
Inferência em dados totais	Funcional	Não	Ontologia	Não
Mapeamento relacional	N/A	Sim (fonte)	Sim (na fonte)	Sim (global)

Soluções de integração baseadas em tecnologia de banco de dados relacional consolidam as fontes por meio de visões GAV. O acoplamento é forte, portanto para editar, adicionar ou remover uma fonte é necessário editar o código da visão, que faz referência a todas as fontes de dados. Dessa maneira, fazer uma modificação em uma das fontes de dados pode instabilizar todo o sistema de integração, o que é indesejável. Além disso, a conciliação semântica depende da construção de funções em linguagem computacional, o que dificulta sua compreensão e validação pelos especialistas no domínio da integração (no caso, medicina).

O sistema FedX transforma uma consulta SPARQL normal em uma consulta federada que utiliza diversos endpoints diferentes e consolida o resultado. Foi criado antes da especificação SPARQL 1.1 e, além da compatibilidade com esse padrão, possui sintaxe própria para a federação. Tem um otimizador e executor de consultas especializado em consultas federadas, mas não implementa inferência sobre a federação (ou seja, cada *endpoint* pode realizar inferência sobre os dados que contém, mas não quando os dados estão em mais de um *endpoint*). Além disso, a falta de inferência sobre a federação requer que a relação entre conhecimento sobre o domínio e sobre o banco de dados seja estabelecida nas fontes.

A Integração de bancos realizada pelo sistema Mastro I realiza inferência em OWL-QL, levando em conta todos os dados da federação, porém requer uma camada extra, de virtualização do banco de dados relacional, para que possa operar. No trabalho referido, é utilizado o IBM Infosphere. Esta camada

extra virtualiza os bancos de dados de origem e implica em maior acoplamento das fontes: quando um novo sistema for adicionado, tanto a camada de virtualização quanto a camada de mapeamento para ontologia deverão ser atualizados.

A metodologia utilizada pelo projeto Debug-IT é híbrida: cada origem de dados deve ser traduzida e materializada num repositório relacional local por meio de ETL, e cada um desses repositórios é então incluído numa federação de bancos de dados relacional, que por sua vez é mapeado em uma camada OBDA utilizando-se o software D2R. O uso de métodos de ETL proporciona maior velocidade no acesso ao repositório local, enquanto não dá acesso aos dados mais recentes de cada fonte. Na federação dos bancos, é utilizado um banco de dados virtual, o que causa uma diminuição na velocidade do acesso e, dada a etapa anterior de ETL, não trará os resultados mais recentes. Esta modelagem, sem vantagens técnicas, é justificada pelos autores como uma medida no sentido de preservar a confidencialidade dos dados de cada centro participante. Entretanto, os dados disponibilizados por esta metodologia não seriam diferentes se houvesse a criação de um repositório estático. Também, toda a harmonização do dados é feita na camada relacional, e apenas depois disto feito é que os dados são traduzidos em RDF, na camada OBDA.

No próximo capítulo, descrevemos uma solução que aborda todos os requisitos apresentados para um sistema de triagem automática de participantes de pesquisa.

4 ESPECIFICAÇÃO E IMPLEMENTAÇÃO

4.1 ANÁLISE DO PROBLEMA

Abaixo analisamos características dos sistemas de informação utilizados em um hospital que impactam na construção de um software para busca de pacientes elegíveis para estudo clínico.

1. Múltiplos sistemas: Em um hospital que cuida do diagnóstico ao tratamento de um paciente, coexistem sistemas de informação especializados em determinados aspectos ou especialidades médicas.

2. Foco dos sistemas não é clínico: Sistemas de informação hospitalar geralmente têm seu foco na gestão do fluxo operacional das clínicas, UTIs, centros cirúrgicos e outras unidades, e na cobrança dos procedimentos junto às operadoras de saúde. A informação clínica é usualmente codificada como texto descritivo. Em alguns casos são utilizados formulários estruturados.

3. Uso de dados recentes: Muitos estudos especificam condições de inclusão ou exclusão que limitam, na prática, a janela de tempo durante a qual um paciente em tratamento é passível de recrutamento. Isso exige que a avaliação do potencial participante de pesquisa considere os dados mais recentes possível.

4. Manutenibilidade: Estando o ambiente de sistemas de informação de um hospital em constante adaptação e evolução, é necessário que a ferramenta de integração e busca acompanhe suas mudanças. Novos sistemas podem ser introduzidos e outros removidos do ambiente de operação, ou o tipo de informação coletada pode ser modificada, exigindo ajustes no sistema de integração. Assim, é importante que um sistema de integração proporcione independência entre as origens de dados.

5. Contextualização de informações: As informações que são registradas em um sistema de informação devem ser interpretadas em função do contexto em que estão inseridas, e isso é fundamental para garantir que o sistema de integração traga resultados corretos.

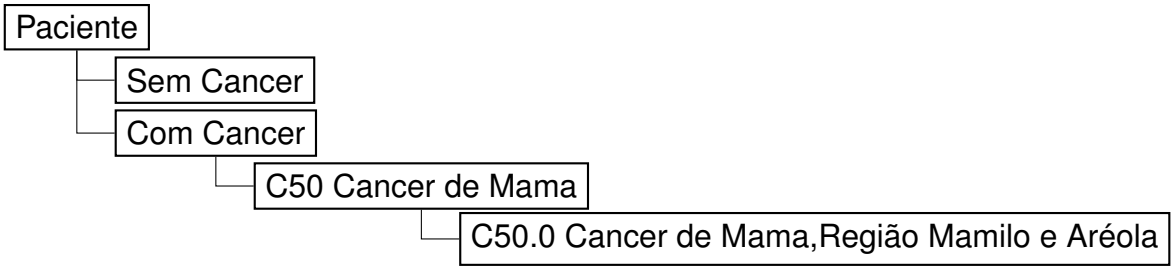

Figura 4 – Hierarquia de conceitos para câncer de mama.

6. Buscar por conceitos relacionados: Uma informação específica sobre um paciente pode significar outras que não estejam explicitamente registradas mas sejam de interesse para o sistema de integração. No exemplo da Figura 4, um paciente com o código CID-10 C50.0 é um paciente de câncer de mama localizado na região do mamilo e aréola, porém é também um paciente com câncer de mama e um paciente com câncer, e é desejável que esse paciente apareça como resultado de busca por qualquer um desses conceitos.

7. Lidar com dados faltantes: Determinada informação pode estar ausente nos sistemas de informação, mas após o exame de informações correlatas o usuário poderá, utilizando conhecimento médico específico, inferir

essa primeira informação. Por exemplo, o paciente com câncer de mama que tenha metástase será classificado, na codificação TNM, como M1. Para câncer de mama, casos M1 tem estadiamento clínico IV, não importando os valores de T ou N. Assim, alguns médicos podem categorizar um paciente apenas com o mínimo de informação necessária para estadiá-lo, ou estadiá-lo sem preencher os valores de T e N.

8. Segregação de conhecimento: Considerando os tópicos acima, vemos que conhecimento técnico sobre os bancos de dados e conhecimento sobre medicina atuarão juntos no sistema de triagem automática. Os profissionais de computação que farão a implementação do sistema raramente são *experts* em medicina, e vice versa. E tratando-se de dois domínios de conhecimento diferentes, é importante que sejam mantidos segregados, de forma que fiquem com acoplamento fraco.

O Quadro 4 resume os problemas discutidos até agora e ferramentas e técnicas para resolvê-las. Em seguida, descreveremos a estrutura do Ontocloud, um software que realiza integração de dados baseada em reescrita de consultas, por meio de ontologias e com capacidade de realização de inferência. Para avaliarmos seu desempenho, comparamos sua performance contra a de dois outros sistemas de integração configurados de maneira equivalente, utilizando como caso de uso um projeto de pesquisa realizado no A. C. Camargo Cancer Center e dados reais de prontuário eletrônico.

4.2 VISÃO GERAL DA ARQUITETURA

Existem três idéias fundamentais por trás da arquitetura do Ontocloud:

- Consultas de usuário têm seus conceitos expandidos de acordo com axiomas previamente definidos;

Quadro 4 – Problemas enfrentados na triagem automática de pacientes e soluções propostas.

#	Problema	Solução
1	Múltiplos sistemas	Integração de dados
2	Foco dos sistemas	Ontologia (harmonização semântica)
3	Acesso a dados recentes	Reescrita de consultas
4	Manutenibilidade	Mapeamento segregado / Ontologia (domínios)
5	Contextualização	Ontologia (anotações)
6	Conceitos relacionados	Ontologia (inferência)
7	Dados faltantes	Ontologia (inferência)
8	Segregação de conhecimento	Ontologia (domínios)

- Ao mesmo tempo, são construídas sentenças que recuperem as informações em cada uma das fontes de dados (*federação*);

- Cada banco de dados relacional é mapeado em uma ontologia por uma camada de reescrita de consultas OBDA, que traduz em tempo real consultas em *SPARQL* para consultas em *SQL*, utilizando-se de consultas *SQL* manualmente elaboradas pelos desenvolvedores da solução de integração.

Mais formalmente, dado um conjunto de bancos de dados $S_{1..n}$, o sistema de integração consistirá em um conjunto de *endpoints SPARQL* $E_{1..n}$ mapeando os objetos da origem na visão global G, utilizando classes de uma ontologia de domínio O_D.

Na Figura 5, cada banco de dados S_i é acessado através de um *endpoint SPARQL* E_i, que deve traduzir as consultas nessa linguagem, utilizando-se da ontologia de domínio, para a linguagem nativa do banco S_i. A partir da consulta original enviada pelo usuário, o módulo de expansão de consultas enriquece a mesma, adicionando termos de acordo com as regras de inferência definidas ao mesmo tempo que indica quais *endpoints SPARQL* cada um dos termos de busca pode ser encontrado. Finalmente, a consulta é executada, os resultados parciais são consolidados e retornados ao usuário.

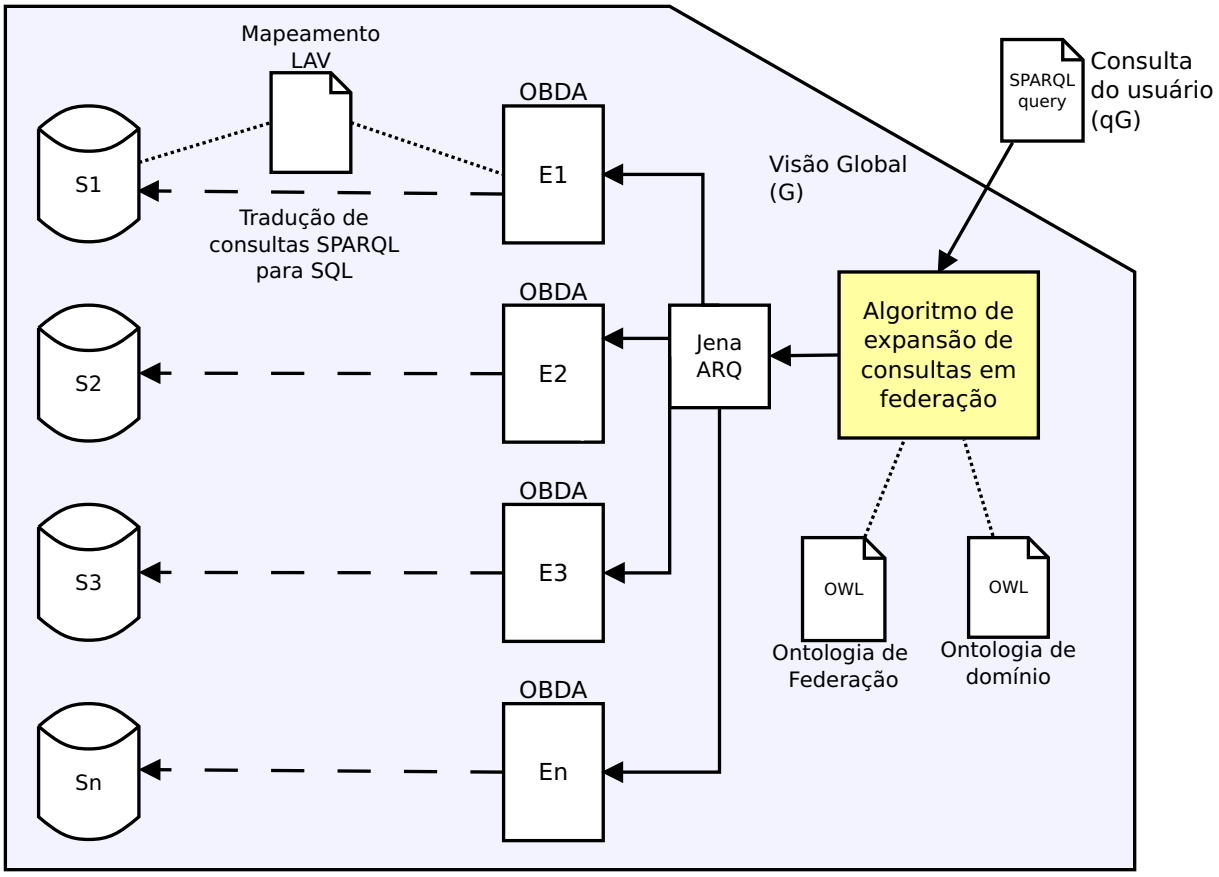

Figura 5 – Arquitetura do sistema Ontocloud.

4.2.1 Ontologias utilizadas

São utilizadas duas diferentes ontologias. A *ontologia de domínio* lista as classes e propriedades que são utilizados para representar a visão global G, assim como suas anotações e relações de inferência. A *ontologia de federação* especifica os bancos de dados de origem quanto ao endereço de acesso e quais classes e propriedades cada qual publica.

A primeira ontologia que deve ser desenhada é a de domínio (O_D), pois é nela que são definidas as classes que correspondem a conceitos que serão buscados, os conceitos que estão descritos nos bancos de dados, e os axiomas que relacionam ambos. Deve ser bem anotada e descritiva, e compreender os conceitos de alto nível que serão consultados, assim como os que estão definidos de fato nos bancos de dados de origem. Esses últimos são representados na ontologia por classes a que chamamos *de base*, pois

são diretamente relacionados com os objetos do banco de dados. As classes que correspondem a conceitos definidos nos termos da consulta são chamadas *classes de consulta*, e devem se relacionar as classes de base por meio de axiomas.

A ontologia de federação (Figura 6) relacionará todos os bancos de dados de origem, o URL de acesso ao *endpoint SPARQL* E_i e quais classes da ontologia de domínio cada um possui mapeamentos. É utilizada pela etapa de expansão e federação de consultas. A ontologia foi criada no *namespace* `http://www.cipe.accamargo.org.br/ontologias/ontocloud2.owl#`[1] e tem as classes e propriedades abaixo:

- `System` (classe): tem como instâncias fontes de dados pertinentes ao problema de integração.
- `Endpoint` (classe): possui como instâncias URLs de *endpoints SPARQL* que serão integrados.
- `systemHasSparqlEndpoint` (propriedade): vincula uma instância de `System` a uma instância de `Endpoint`, significando que uma fonte de dado tem seus dados mapeados num determinado *endpoint SPARQL*.
- `systemImplementsClass` (propriedade): vincula uma instância de `System` a uma classe, significando que aquela fonte de dados possui mapeamentos para a classe especificada.

Essa última propriedade relaciona uma instância a uma classe por uma propriedade criada na própria ontologia; isto torna a ontologia representável apenas por OWL-FULL, que não é decidível[2]. Entretanto, como não é necessário o uso de regras de inferência além do estabelecido pelo RDFS, esse

[1] Utilizamos a padronização de colocar os nomes de classe com a primeira letra em maiúscula, e de propriedades em minúsculas

[2] Tabela 10 em `http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/`

nível de expressividade não é um problema, não importando o número de instâncias e classes da ontologia de domínio que se utilizem.

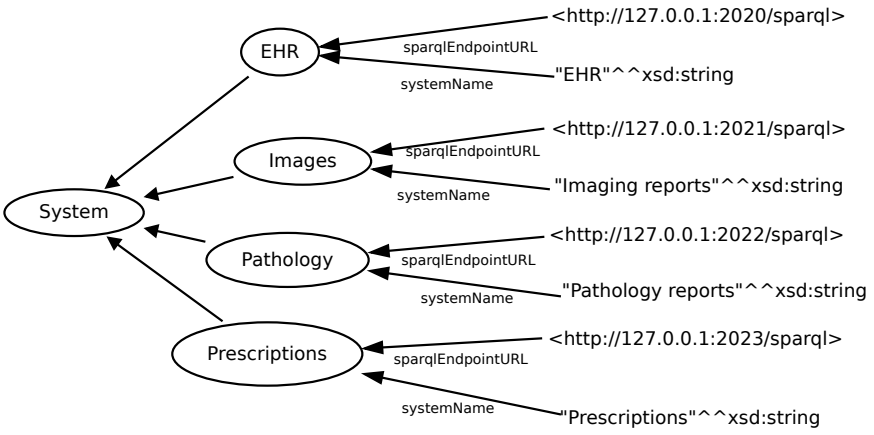

Figura 6 – Exemplo de ontologia de federação.

4.2.2 Mapeamento

Para relacionar os objetos do banco de dados de origem com os da ontologia global, deverão ser construídos *mapeamentos*. Podem ser LAV ou GAV, desde que não acoplem de maneira forte as fontes de dados. Serão utilizados pela ferramenta de OBDA, que traduzirá consultas q_E sobre a ontologia de domínio para consultas em SQL usando as definições desse banco de dados de origem em específico.

4.3 PROCESSO DE CONSTRUÇÃO DAS ONTOLOGIAS

Abaixo uma proposta de metodologia para construção das ontologias, quando da construção de um sistema de integração de dados utilizando-se o Ontocloud.

1. Elaborar uma lista descrevendo as questões de competência que o sistema de integração deverá abordar, com exemplos de consultas que deverão ser respondidas, e exemplos de respostas corretas que podem ser obtidos;

2. Identificar conceitos-chave nessas questões, e representá-las como classes na ontologia de domínio O_D; anotar nas classes o significado do termo e em qual questão de competência surgiu; serão chamadas de *classes de consulta*;

3. Descrever os exemplos de consulta em linguagem *SPARQL*, utilizando-se das classes de consulta;

4. Identificar, dentre os bancos de dados que compõe o ambiente de sistemas de informação relevante, quais deles contém dados sobre os conceitos buscados;

5. Para cada classe identificada nas questões de competência, localizar nos bancos de dados as tabelas e colunas que possuem informação com semântica mais próxima; é fundamental a participação de especialistas em banco de dados e usuários experientes; adicionar à ontologia de domínio a classe, anotando o banco de origem e semântica exata; as classes criadas nesse ítem serão chamadas *classes de base*; criar na ontologia de federação uma instância correspondente a esse banco de dados, e relacioná-la com as classes de base que implementa.

6. Relacionar as classes de base às classes de consulta por meio de axiomas da ontologia;

7. Construir consultas SQL nas fontes de dados correspondentes a cada uma das classes de base; colocar essas consultas na ontologia de mapeamento relativo àquele banco de dados;

8. Executar as consultas *SPARQL* correspondente às consultas de exemplo, e verificar sua precisão e cobertura em relação aos resultados especificados;

4.4 ALGORITMO DE EXPANSÃO DE CONSULTAS

Nessa seção será descrito o algoritmo que realiza a expansão de consultas em federação, responsável pela integração dos dados e inferência. Também serão descritas estratégias de otimização dessas consultas e de planejamento de execução, com o objetivo de melhorar a performance das consultas.

4.4.1 Federação de consultas

Em uma federação composta por diversas fontes de dados que contém informação sobre instâncias de uma classe C, para obter a listagem completa dessas instâncias, é preciso consultar cada uma das fontes e unificar os resultados. De maneira formal, sejam fontes de dados $S_{1..n}$ mapeadas em classes de uma ontologia de domínio O_D em *endpoints SPARQL* $E_{1..n}$:

$$q_G^C \equiv \bigcup q_{E_i}^C \{\forall E_i | implementaClasse(E_i, C)\}$$

Onde q_G^C é uma consulta à visão global G por instâncias da classe C, e $q_{E_i}^C$ é uma consulta ao *endpoint SPARQL* E_i por instâncias da classe C. A transformação de consulta *SPARQL* pode ser ilustrada da seguinte maneira:

```
SELECT * { ?x a C }  →  SELECT * {
                            SERVICE(<E1>) { ?x a C }
                            UNION SERVICE(<E2>) { ?x a C }
                             ...
                            UNION SERVICE(<En>) { ?x a C }
                        }
```

No caso de alguma das classes utilizadas na consulta participar de axiomas de inferência, e os componentes desses axiomas estarem contidos em

algum *endpoint*, podemos substituir a referência à classe original por referências às subclasses dessa. Assim a inferência é implementada no algoritmo de expansão de consultas e leverá em conta os dados contidos na federação inteira.

4.4.2 Inferência em consultas

Suponha que A,B,C e D sejam classes relacionadas entre si pelos axiomas:

$$D \sqsubseteq C$$

$$A \sqcap B \sqsubseteq C$$

O primeiro axioma define que D é *subclasse* de C, ou seja, toda instância de D também é instância de C. Pelo segundo axioma, a *intersecção* das classes A e B é subclasse de C (ou, se uma instância x pertence às classes A e B simultaneamente, então segue que ela pertence à classe C). Uma consulta pelas instâncias da classe C portanto pode ser expandida como uma consulta às instâncias que pertencem a A e B, mais as instâncias que pertencem a D. Assim, temos a seguinte expansão de consulta:

```
SELECT * { ?x a C }  →  SELECT * {
                            { ?x a C }
                            UNION { { ?x a A } { ?x a B } }
                            UNION { ?x a D }
                        }
```

Devemos unir os resultados de $D \sqcup A \sqcap B$ aos próprios resultados de C; o fato dessa classe poder ser populada por inferência não elimina a possibilidade de algumas instâncias estarem explicitamente vinculadas a ela. Na

Figura 7, exemplificamos a aplicação desse algoritmo de inferência em consultas.

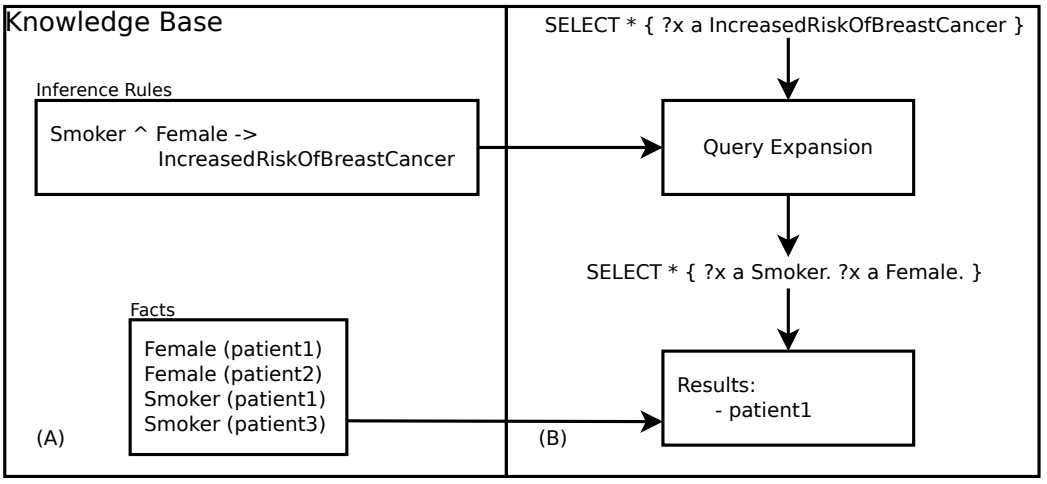

Figura 7 – Exemplo de aplicação do algoritmo.

Dessa maneira, implementamos a semântica de `rdf:subClassOf` [3] e `owl:intersectionOf` [4] em *SPARQL*. Estas duas regras estão contidas em OWL-EL, que tem inferência decidível. Isto garante que as consultas sempre poderão ser reescritas.

4.4.3 O Algoritmo

Os dois processos descritos acima, de federação e inferência em consultas, devem ser combinados para que as consultas possam integrar os dados da federação e realizar inferência nesses dados. A idéia principal consiste em aplicar as transformações de inferência descritas em 4.4.2 à federação de consultas de 4.4.1. No Quadro 5 é apresentado o algoritmo que realiza essa tarefa. Ele retorna uma árvore sintática da consulta *SPARQL*, para posterior compilação em linguagem de consulta.

[3] regras RDFS9 e RDFS11 definidas em `http://www.w3.org/TR/rdf-mt/`

[4] `http://www.w3.org/TR/owl-semantics/rdfs.html#5.2`

Quadro 5 – O Algoritmo

```
SPARQLFederator ( class, O_D, O_F )
Q ← empty query syntactic node ;
E ← O_F.getEndpointsThatImplements(class) ;
if not E.empty() then
    while e ← E.pop() do
        Q ← new UnionNode( new ServiceNode( e, class ), Q ) ;
    end
end
S ← O_D.getSubClasses(class) ;
if not S.empty() then
    while s ← S.pop() do
        if s.isClass() then
            Q ← new UnionNode( SPARQLFederator( sc, O_D, O_F)
            , Q ) ;
        end
        if s.isObjectIntersectionOf() then
            SC[] ← s.getClasses() ;
            while sc ← SC.pop() do
                Q ← new JoinNode( SPARQLFederator( sc, O_D,
                O_F), Q ) ;
            end
        end
    end
return Q
```

4.5 OTIMIZAÇÃO E PLANEJAMENTO

É comum que os mecanismos de execução de consultas incluam etapas de otimização (redução de trabalho redundante) e planejamento (escolha da ordem de operações mais eficiente) antes da resolução propriamente dita da consulta. Não é objetivo deste trabalho tornar a implementação de *Ontocloud* a mais eficiente possível, porém algumas otimizações simples e um protótipo de algoritmo de planejamento foram implementadas, e estão descritos no apêndice 7.

4.6 IMPLEMENTAÇÃO

A implementação do algoritmo foi feita utilizando-se a linguagem Java 1.6[5]. Utilizamos a biblioteca Jena 2.10.1[6] para lidar com consultas *SPARQL*, arquivos OWL e execução de consultas. Para testes automatizados, utilizamos JUnit 3.8.1[7]. O software foi publicado com licença de código aberto no repositório GitHub[8], e pode ser encontrado no URL abaixo:

`https://github.com/djogopatrao/SPARQLFederator`

4.7 ANÁLISE CRÍTICA DO ONTOCLOUD

Uma vez desenhado e implementado o sistema Ontocloud, atualizamos o Quadro 3, colocando como última coluna o sistema Ontocloud para comparação com os outros sistemas avaliados no Quadro 6.

Quadro 6 – Ontocloud e demais sistemas de integração avaliados.

Sistema	Relacional	FedX	Mastro-I	DebugIT	Ontocloud
Acesso dinâmico	Sim	Sim	Sim	Não	Sim
Acoplamento do mapeamento	Forte	N/A	Fraco	Forte	Fraco
Reconciliação semântica	Funcional	Ontologia	Ontologia	Funcional	Ontologia
Camada de integração	BDR	Ontologia	Ontologia+BDR	BDR	Ontologia
Inferência em dados totais	Funcional	Não	Ontologia	Não	Ontologia
Mapeamento relacional	N/A	Sim (fonte)	Sim (fonte)	Sim (global)	Sim (fonte)

Na Figura 8 vemos uma comparação estrutural entre os métodos DebugIT, Mastro-I e Ontocloud. DebugIT realiza a integração de dados de forma estática (*data warehouse* ou ETL) em camadas relacionais e apenas então publica os resultados em um *endpoint SPARQL*. Mastro-I realiza a integração de forma dinâmica e após a tradução dos dados relacionais para ontologia, mas requer uma camada relacional para consolidar as fontes de dados, chamada *Virtual Data Base* (VDB). Ontocloud não depende de soluções relacionais,

[5] `https://www.java.com/`
[6] `http://jena.apache.org/`
[7] `http://junit.org/`
[8] `https://github.com/`

pois a integração é feita sobre a camada OBDA, por meio de expansão de consultas SPARQL.

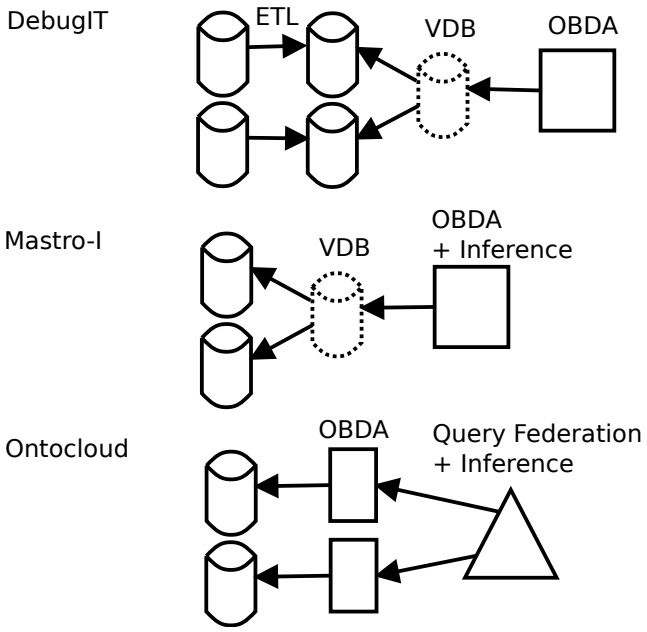

Figura 8 – Comparação de algumas soluções de integração de dados por ontologias.

O principal diferencial com relação ao FedX, que também realiza expansão de consultas para integrar os resultados de toda a Federação, é a possibilidade de realizar inferência sobre os dados de toda a Federação. No FedX, ao se realizar a consulta, cada uma das fontes de dados é consultada sobre exatamente as propriedades e classes especificadas na consulta. No Ontocloud, não apenas as classes diretamente especificadas podem ser consultadas, mas também as que possibilitem a inferência da classe consultada, como ilustra a Figura 9.

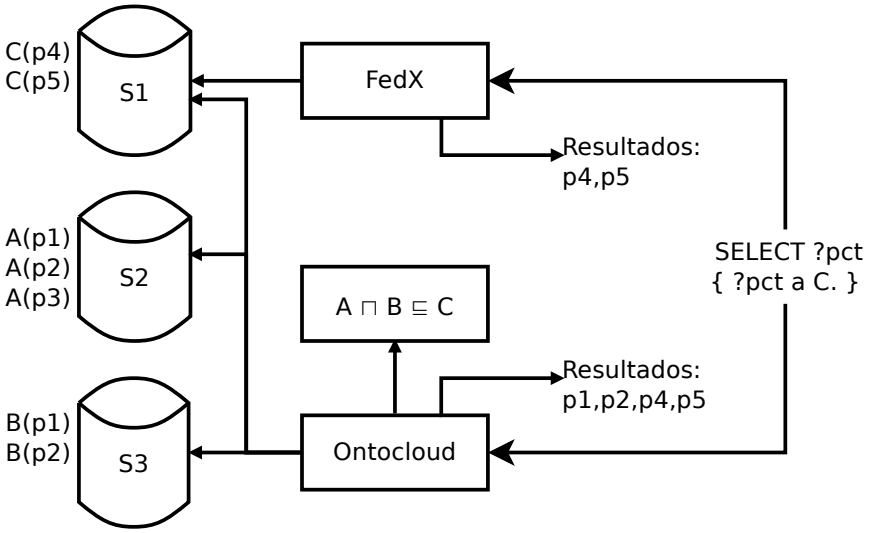

Figura 9 – Federação de consultas do FedX e Ontocloud.

5 AVALIAÇÃO

Para verificar se o sistema proposto atingirá os objetivos desejados, elaboramos um experimento cujo objetivo é simular a seleção de pacientes para um estudo clínico efetivamente executado no A. C. Camargo Cancer Center, utilizando dados reais nos bancos de dados dessa instituição. O objetivo é encontrar os pacientes incluídos de fato no estudo e outros que também atendam aos critérios de seleção estipulados.

Para avaliar características técnicas do sistema, compararemos os resultados obtidos com aqueles de outros dois sistemas de integração: Um baseado em ontologias e acesso estático, para avaliar o algoritmo de expansão de consultas federadas, e outro baseado em bancos relacionais e acesso dinâmico, para avaliar o tempo de resposta. Avaliaremos também se os requisitos funcionais do sistema, integração de bancos de dados, reconciliação semântica, obtenção de dados recentes, independência entre fontes e separação entre conhecimento médico e sobre o banco de dados (estabelecidos no capítulo 4.1) foram atingidos.

5.1 FUNDAMENTOS MÉDICOS E BIOLÓGICOS RELEVANTES

O câncer de mama é uma doença que afeta principalmente mulheres com mais de 40 anos de idade; pacientes com menos dessa idade representam 5% do total de pacientes com câncer de mama (Sariego 2010). Os tipos mais comuns de câncer de mama são os carcinomas ductais, que originam-se nos ductos mamários, e os lobulares, nos lóbulos. O tratamento é cirúrgico, podendo haver quimioterapia neoadjuvante e/ou adjuvante (ou seja, anterior à

cirurgia, com o intuito de diminuir o tumor, ou posterior, para eliminar células cancerosas remanescentes), além de hormonioterapia. As drogas anti carcinogências mais comuns em uso atualmente são terapias de bloqueamento hormonal, que agem em tumores que exigem a presença de hormônios como o estrogênio para continuar seu crescimento, quimioterapia, que destroem células em processo de multiplicação, e anticorpos monoclonais, que ligam-se a outros participantes do ciclo celular para reduzir a velocidade de multiplicação ou impedir processos como a angiogênese (Florescu et al. 2011) .

Os tumores de mama podem ser classificados quanto à superexpressão da proteína erbB-2, codificada pelo gene HER2 (ou ERBB2). Essa proteína é composta por um domínio de ligação extracelular e um intracelular. Quando um ligante adequado acopla-se ao seu domínio extracelular, inicia-se uma cascata de reações que dispara o mecanismo de multiplicação celular e inibe a apoptose. 20-30% dos tumores de mama apresentam amplificação do HER2 e consequentemente superexpressão da erbB-2 (Mitri et al. 2012). Tumores de mama com essa característica tem associação com maior probabilidade de recorrência e pior prognóstico (Tan e Yu 2007). O diagnóstico de um tumor HER2 positivo depende da avaliação por imunohistoquímica de uma seção do tumor; caso o patologista avalie como três cruzes (ou seja, muito forte) a presença do receptor, o tumor é considerado HER2 positivo. Porém, se a avaliação for duas cruzes (forte), é preciso executar a avaliação do marcador por FISH ou CISH; caso a relação entre o número de cópias do HER2 e do cromossomo 17 seja superior a 2.2, então o tumor é considerado HER2 positivo.

Existem drogas especialmente desenhadas para bloquear o mecanismo de proliferação mediado pelo erbB-2. O trastuzumab é uma molécula que liga-se ao domínio extracelular dessa proteína, inibindo a proliferação celular, por isso é categorizado como uma terapia "anti-erb". Em casos já metastatizados, mostrou-se que essa droga aumentou a sobrevida média dos pacientes de

20.3 a 25.1 meses (Hudis 2007). Já para cânceres em fase inicial, reduz o risco de recorrência em 9.5% (Moja et al. 2012). É comum a combinação de trastuzumab com agentes anti-neoplásicos (Nahta e Esteva 2003).

5.2 ESTUDO CLÍNICO GLICO-801

O estudo de fase 2 GLICO-0801 (GLICO - Grupo Latino Americano de Investigacoes Clinicas em Oncologia 2014) foi realizado no A. C. Camargo Cancer Center, dentre 18 centros na América Latina, entre 2009 e 2011. Seu objetivo era o de avaliar a segurança de três tratamentos baseados na droga Lapatinib (combinado com Capecitabina, Gemcitabina ou Vinorelbina) em pacientes com adenocarcinoma invasivo de mama HER2+, com tratamento de primeira linha metastática com taxanos, ou tratamento adjuvante ou neo adjuvante com esse tipo de drogas. Na época, o tratamento padrão, aceito em mais de 20 países, para tumores que não respondem ao tratamento com taxanos, era a combinação Lapatinib e Capecitabina.

O processo que foi seguido nesse estudo para encontrar pacientes dependia dos médicos participantes identificarem, no momento do atendimento de um paciente, se o mesmo atendia os critérios principais do estudo (câncer de mama HER2+ avançado ou metastático), apresentar o estudo e convidá-lo a realizar outros testes para ingressar no mesmo. Em 6 meses de janela de recrutamento, 7 pacientes foram identificados e recrutados no A. C. Camargo Cancer Center. Não houve registro dos pacientes que atendiam aos critérios de seleção mas não puderam ser recrutados, por uma ou outra razão.

5.3 CONCEITOS RELEVANTES PARA O CASO DE USO

Obtivemos os critérios de inclusão e exclusão no site `http://clinicaltrials.gov`. Estudamos esses critérios e classificamos cada um deles como de Seleção ou Recrutamento, onde os primeiros podem ser avaliados usando-se informações do prontuário eletrônico, e os últimos necessitam de procedimentos e testes específicos para o projeto. Abaixo listamos os critérios considerados (com a numeração utilizada no estudo).

- Adenocarcinoma mamário invasivo (critério 3)
- Doença metastática (Estádio IV) ou avançada (Estádio IIIb ou IIIc com lesão T4) (critério 3)
- Superexpressão de HER2 confirmado por imunoistoquímica e/ou hibridação fluorescente in situ (critério 6)
- Passou por tratamentos quimioterápicos prévios (histórico de terapia com taxanos) (critério 7)

Verificamos que três sistemas em uso atual no A.C. Camargo Cancer Center continham dados relevantes para essa busca.

- *EHR*: Sistema de gestão hospitalar. Contém prontuário eletrônico, com formulários preenchidos pelo médico no momento da consulta ambulatorial, diagnósticos e prescrição de medicamentos infundidos na instituição, em especial os quimioterápicos.
- *AP*: Sistema de gestão de laboratório de anatomia patológica. Armazena os laudos de anatomia patológica, incluindo macroscopia, microscopia, imunohistoquímica e outros testes diagnósticos.

- *RHC* (Registro Hospitalar de Câncer): Sendo o câncer uma doença de notificação obrigatória, há um departamento no A. C. Camargo Cancer Center que registra todos os casos diagnosticados e tratados nesse centro e os acompanha, capturando informações anualmente.

Abaixo descrevemos, para cada um dos conceitos referidos pelo critério de seleção que escolhemos, os conceitos mais próximos que puderam ser encontrados nos bancos de dados contemplados. Um resumo encontra-se no Quadro 7.

- *Câncer de Mama*: O diagnóstico pode ser encontrado em praticamente todos os documentos do prontuário eletrônico (sistema EHR) no campo CID-10. O código correspondente ao câncer de mama (incluindo o adenocarcinoma invasivo e outros tipos) é C50, compreendendo os códigos de C50.0 a C50.9. Essa informação também foi encontrada no banco de dados do sistema RHC, codificado da mesma maneira. No sistema AP a informação está presente em campo texto, porém de maneira indireta: na seção de macroscopia, a descrição da peça recebida para o exame inclui a localização da peça extraída. Para concluir que trata-se de um câncer de mama, é preciso combinar a localização com o diagnóstico patológico.

- *Adenocarcinoma Invasivo*: A descrição do subtipo histológico, "adenocarcinoma invasivo", foi encontrada em dois tipos de documentos, nos Laudos de anatomia patológica (sistema AP) e no sistema RHC. No primeiro, em textos semiestruturados (ou seja, são textos livres com ordem padronizada) e no último, estruturados e codificados com o CID-O 3a. edição. Nessa codificação, os adenocarcinomas invasivos são classificados em códigos entre 8140/X e 8389/X, sendo que X pode ser 3 (invasivo) ou 6 (metastático). No sistema EHR, a informação pode estar em

documentos de prontuário, mas foi identificado que é comum utilizar-se de abreviações ("CDI"para Carcinoma Ductal Invasivo) e nem sempre a informação está registrada.

- *HER2+*: O conceito HER2Positivo pode ser deduzido a partir do laudo de imunoistoquímica no sistema AP, caso esse traga um resultado com escore 3, ou com escore 2 e mais o resultado de FISH ou CISH indicando uma relação entre o número de cópias do gene HER e o cromossomo 17 maior que 2,2. No sistema EHR, essa informação pode ou não ser encontrada de forma discursiva em diversos campos diferentes, porém sem detalhamento de origem da informação. No sistema RHC não é registrada essa informação.

- *Doença Metastática ou Localmente Avançada*: A definição desses conceitos pode ser encontrada na própria descrição do estudo, que os define em termos do estadiamento TNM 6a edição. Nos documentos eletrônicos do sistema EHR pudemos encontrar campos correspondentes a esses estadios T,N,M implementados em quatro documentos de prontuário, de formas diferentes: como caixas de seleção, como checkboxes ou caixas de texto livre. No sistema RHC, essa informação está disponível em campos estruturados. No sistema AP, essa informação pode ser inferida após a análise da seção Macroscopia sobre o tumor, os linfonodos afetados e/ou a metástase à distância tem o tamanho e características descritos, e na conclusão do laudo sobre a malignidade ou não da lesão - note-se que nesse laudo apenas consta informação sobre metástase se a peça metastatizada estiver em análise.

- *Tratamento com taxanos*: Os tratamentos quimioterápicos são prescritos de maneira estruturada através do sistema EHR pelos médicos do

departamento de Oncologia Clínica. Existe um código de ítem de prescrição para cada um dos diferentes medicamentos prescritos, em particular, para os medicamentos de interesse (taxanos), nome comercial ou dosagem diferentes implicam em códigos diferentes (não há como distinguir o princípio ativo). No banco de dados do sistema RHC, há o registro apenas se o paciente passou por quimioterapia ou não, sem especificar o medicamento, dose e duração do tratamento. O sistema AP não contempla esse tipo de informação.

Quadro 7 – Conceitos de critério de seleção e implementações em banco de dados de produção.

Critério	Sistema	Conceito BD	Estruturado?	Comentário
Adenocarcinoma mamário invasivo (critério 3)	EHR	C50.*	S	Vários documentos
	AP	Morfologia em texto	N	Topografia está em descrição da peça
	RHC	M8140:8389, C50	S	Topografia e morfologia
HER2+	AP	Em texto	N	HER2: [0-3]+/positivo ou Relação Cópias/CR17
	EHR	Em texto	N	Não há detalhes ou exame de origem
	RHC	Não mapeado		
Doença metastática	EHR	M1	S	Vários documentos
	RHC	M1	S	Campos TNM
	AP	Em texto	N	Apenas se peça analisada for uma metástase
Doença localmente avançada	EHR	Definição TNM	S	Documento da oncologia clínica
	RHC	Definição TNM	S	Campos TNM
	AP	Em texto	N	Nem sempre está descrito
Uso de taxanos	AP	Não há		
	EHR	Medicamento prescrito: Taxol, Docetaxel, Paclitaxel	S	Prescrições
	RHC	Quimioterapia S/N	S	Sem detalhes
Maior de 18 anos	AP	Data de nascimento	S	Metadados de paciente
	RHC	Data de nascimento	S	Tabela de registro
	EHR	Data de nascimento	S	Cadastro do paciente

5.3.1 Qualidade

A qualidade das informações contidas nos bancos de dados apresentados para a realização de busca carece de avaliação objetiva adequada. Como exceção, um relatório interno do Laboratório de Informática Médica do AC Camargo realizou uma avaliação de precisão da obtenção de pacientes com diagnóstico positivo para câncer e chegou ao valor de 96%.

Algumas informações, pela natureza de seu processo de coleta, podem ser consideradas de boa qualidade. O banco de dados RHC é construído manualmente pelo departamento de Registro Hospitalar de Cancer. Os

prontuários são consultados pelos colaboradores desse setor, que preenchem campos estruturados, dentre eles o diagnóstico topográfico e morfológico codificado sob o CID-O. O banco de dados EHR contempla um amplo espectro de informações, que seguem padronizações diferentes, o que se reflete na qualidade. Por exemplo, as informações sobre prescrições de medicamentos para pacientes internados ou quimioterápicos são registradas pelos médicos. Depois as enfermeiras realizam o aprazamento, ou determinação do horário em que cada medicação será infundida, os farmacêuticos fazem a preparação dos medicamentos que requeiram isto (como alguns quimioterápicos), e por fim estas informações são utilizadas para realizar cobrança. A informação de prescrição, por passar por muitas etapas de conferência, englobando diferentes profissionais, também pode ser considerada fidedigna. Outras informações, como o estadiamento ou mesmo o diagnóstico morfológico do paciente, são cadastradas em campo texto junto com outras informações, o que significa que não tem codificação, padronização ou mesmo preenchimento obrigatório.

As informações dos laudos de anatomia patológica (AP) seguem um texto padronizado para cada topografia ou morfologia. Com o passar dos anos, o texto padrão foi alterado, não de maneira que impeça o cadastro ou leitura da informação por pessoas, mas dificultando a obtenção automática por um programa de computador. O diagnóstico patológico, informação central na composição do laudo, por vezes é demasiadamente genérico e nunca é codificado, sempre constando o termo preferido pelo patologista. O status do marcador HER2 é alvo de avaliação direta do patologista, porém a informação está em formato bruto, ou seja, contém a descrição da observação, seja pelo método de imunohistoquímica ou por imunofluorescência (FISH ou CISH). Assim, os subsídios para concluir se determinado tumor é ou não HER2 positivo estão disponíveis para o leitor, porém complicam a obtenção automática desta informação.

5.4 CRIAÇÃO DO MAPEAMENTO

Para cada uma das classes diretamente representadas pelos bancos de dados, escrevemos uma consulta SQL que retorna o identificador dos pacientes que atendem a esse critério. Uma consulta SQL foi escrita para cada conceito em cada banco de dados. Especificamente no caso do estadiamento TNM, que no banco EHR é definido de maneiras diferentes, foi necessário criar mapeamentos diferentes dentro do mesmo banco. No Anexo A descrevemos as consultas utilizadas.

5.5 CRIAÇÃO DAS ONTOLOGIAS

Representamos como classes OWL os conceitos descritos tanto pelos critérios de inclusão quanto aqueles encontrados no banco de dados na ontologia de domínio. Essas classes foram incluídas na ontologia de domínio. Vinculamos, por meio de axiomas de subsunção e interseção, as classes pertinentes ao critério de seleção às relacionadas à informação representada nos bancos de dados.

Criamos a ontologia de federação, gerando instâncias da classe `System` para cada um dos sistemas (EHR, AP e RHC) que serão integrados. Vinculamos cada uma dessas instância| às classes que nele são representadas por meio da propriedade `systemImplementsClass`. A propriedade `systemHasSparqlEndpoint` foi definida para cada sistema mais à frente, na etapa experimental.

No capítulo 4.4.2, restringimos a inferência de nosso algoritmo de expansão de consultas a duas regras, a que implementa subclasses e a que implementa intersecções de classes. Para a aplicação abordada neste capítulo, isto será suficiente. Por exemplo, podemos dizer que o código C50.1 é

uma subclasse do código C50, ou que um paciente estadiado com T4 e M0 e diagnosticado com C50 tem estádio clínico ECIIIb.

A figura 10 representa a ontologia criada. A relação `is-a` representa a relação de subclasse. As regras envolvendo intersecção de classes são discutidas ao longo do capítulo e estão todas no Anexo A.

A figura 11 ilustra a ontologia de federação, com bancos de dados de origem (com os nomes descritos acima) e algumas das classes que cada um pode mapear. Foi construída com base nos dados obtidos no Quadro 7.

Assim, vemos que os bancos de dados mapeiam conceitos diferentes que estão relacionados entre si. Um exemplo é o conceito “Quimioterapia”, mapeado pelo banco Registro Hospitalar, e o tipo do medicamento (por exemplo, Docetaxel), mapeado pelo banco “Prescricao”. Apesar de uma busca pelo medicamento específico não retornar pacientes que estão mapeados apenas no conceito Quimioterapia, a busca pelo conceito mais genérico retornará todas as instâncias que estão vinculadas hierarquicamente a ele, incluindo os medicamentos mais específicos.

Para encontrarmos os pacientes elegíveis, criamos uma classe `CritérioTriagem` que, por meio dos axiomas abaixo, engloba os pacientes que atendam a todos os critérios especificados (Quadro 8).

Quadro 8 – Axiomas que descrevem os critérios de inclusão utilizados.

$$DoencaLocalmenteAvancada \sqcap HER2positivo \sqcap CancerDeMama \sqcap Taxanos \sqcap AdenocarcinomaInvasivo \sqsubseteq CriterioTriagem$$
$$DoencaMetastatica \sqcap HER2positivo \sqcap CancerDeMama \sqcap Taxanos \sqcap AdenocarcinomaInvasivo \sqsubseteq CriterioTriagem$$

Figura 10 – Taxonomia de classes da ontologia de domínio.

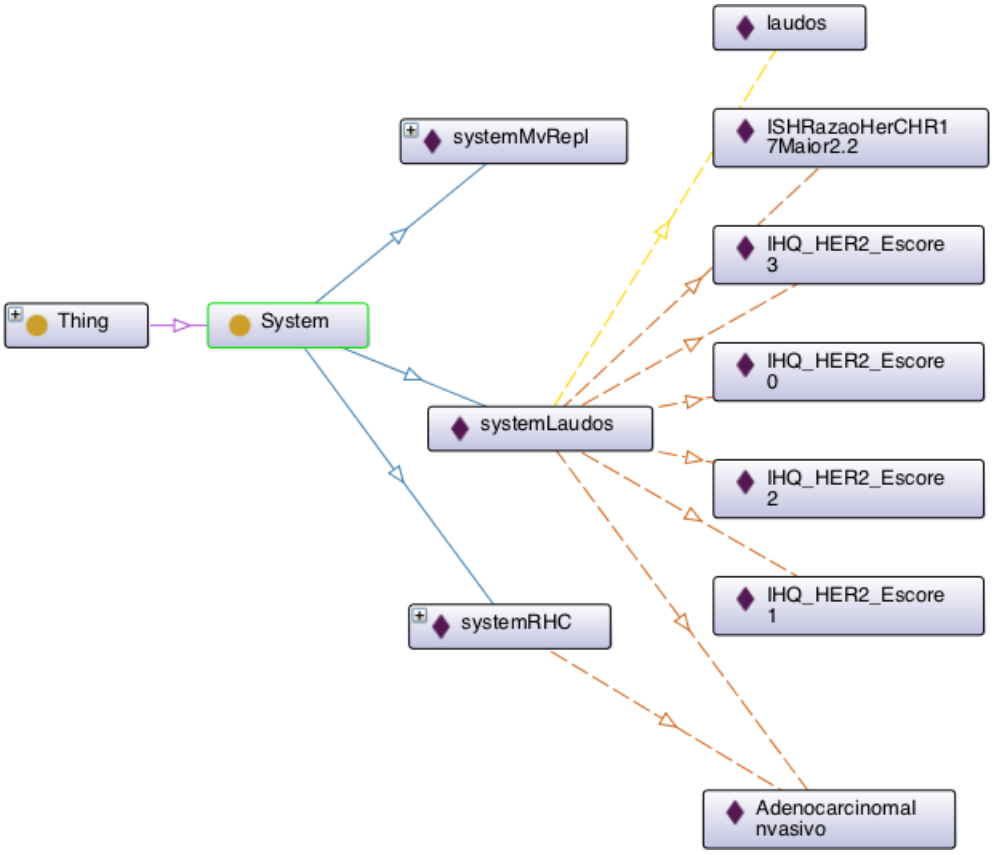

Figura 11 – Excerto da ontologia de federação criada para a aplicação.

5.6 MAPEAMENTO DE DIAGNÓSTICO DE CÂNCER DE MAMA

A informação sobre a topografia (órgão) do câncer está presente nos três sistemas, porém em contextos diferentes que merecem discussão. No sistema RHC, o campo é explicitamente definido como topografia e aceita a codificação topográfica definida no CID-O 3a. edição (a qual é ligeiramente diferente dos códigos CID-10). No sistema EHR, existem diversas tabelas e contextos diferentes onde o CID-10 é coletado: atendimento ambulatorial, internação, prescrição, cirurgia, e documentos eletrônicos, que podem corresponder às mais diversas situações como campanhas de prevenção e resumo de óbito. Escolhemos a tabela que traz informação sobre atendimento ambulatorial e pacientes internados, pois tem maior abrangência e recebem os valores preenchidos diretamente pelos médicos. Porém, não há garantia de

que um paciente que tenha o código relativo a câncer de mama atribuído seja de fato um caso confirmado de câncer - o código é atribuído também a pacientes com suspeita desse tumor, tendo assim a função de hipótese diagnóstica. O código utilizado nessa tabela é o CID-10. No sistema AP, a informação topográfica geralmente está descrita na seção de exame macroscópico da peça de forma textual, assim como na conclusão é descrito se é um tumor maligno ou não e o tipo histológico do mesmo.

Decidimos utilizar as informações contidas nos dois primeiros bancos, o EHR e o RHC, por esses conterem essa informação já codificada (apesar dos códigos CID-10 e CID-O, a rigor, serem diferentes, em particular no caso de câncer de mama são o mesmo, C50). Assim, evitamos o problema de interpretar texto corrido no sistema AP.

5.6.1 Adenocarcinoma Invasivo

Esse é um conceito relacionado à morfologia e comportamento da célula tumoral. O mesmo é um câncer originário de células glandulares que potencialmente pode invadir tecidos adjacentes ou órgãos à distância (metástase). Está presente nos três sistemas contemplados, de maneiras e em locais diferentes. No sistema EHR, não há campo específico para o diagnóstico anatomopatológico do tumor; em nossa análise, verificamos que esses dados são cadastrados em formato textual em diversos documentos e campos diferentes. Já no sistema AP, também está em formato textual, mas diferentemente do sistema EHR, todo o texto do documento está em um único campo, facilitando a consulta. No sistema RHC, o diagnóstico é cadastrado de forma estruturada com o código morfológico do CID-O 3a. edição. Também, essa informação é o tema principal do laudo anatomopatológico, sendo que no EHR é uma informação que pode ou não constar do documento. Assim, decidimos por utilizar a informação constante nos sistemas AP e RHC.

Adenocarcinoma é todo câncer que se origina em tecido glandular.

Sendo câncer, tem potencial metastático, portanto o adjetivo “invasivo”, aplicado a esse substantivo, é redundante. Consideramos portanto qualquer laudo contendo a palavra “adenocarcinoma” como indicativo de que se trata de um câncer desse tipo. Incluímos também os subtipos de adenocarcinoma mamário “carcinoma tubular”, “carcinoma ductal” e “carcinoma cribriforme”.

5.6.2 Mapeamentos para HER2+

Criamos os conceitos `RelacaoHER2CR17SuperiorA2.2`, `HER2IHQ++` e `HER2IHQ+++` para indicar o que efetivamente está representado nos textos de laudos, e criamos consultas SQL utilizando expressões regulares que identificam os laudos que atendem a esse critério. A partir de uma expressão inicial (HER2 ou CERBB2) estudamos o texto dos laudos para identificar aqueles que indiquem essas condições, e quais sequencias de caracteres são recorrentes nesses laudos. Com essas sequencias recorrentes, criamos expressões regulares mais complexas para refinar o resultado (retornando apenas aqueles que indiquem exatamente a condição desejada).

Uma outra representação dessa informação seria por meio de propriedades de dados. O score HER2/CR17 poderia ser atribuído a uma propriedade de dados (por exemplo, diríamos `paciente1 temRazãoHER2CR17 3.2`), e por meio de axiomas, definiríamos a regra (> 2.2) para considerar um cancer HER2+. Essa abordagem, do ponto de vista de separação entre conhecimento médico e de banco de dados, é mais adequada aos nossos propósitos, pois a definição desse critério de corte é puramente patológica, e na nossa implementação, ficou definida no domínio do banco de dados. Atualmente, o sistema de expansão de consultas federadas não contempla propriedades de dados e restrições de valores. Fizemos, portanto, uma concessão da separação de domínios de conhecimento em favor da performance e da inclusão dessa importante variável.

5.6.3 Diferentes representações do estadiamento TNM

No caso específico do estadiamento TNM, são 4 os documentos que contém essa informação no sistema EHR, e são os mais preenchidos: Admissão Radioterapia: O TNM é apresentado em campos checkbox, com uma opção para cada categoria:

- TX, T1, T2, T3, T4
- NX, N0, N1, N2, N3
- MX, M0, M1

Ou seja, cada um dos campos acima (agrupados artificialmente) recebe um valor "marcado" ou "não". Apesar desses campos permitem maior agilidade ao médico, que tem todas as categorias à vista, não é possível impedir que duas categorias concorrentes (como T1 e T4) sejam simultaneamente marcadas, o que pode levar a inconsistências. Além disso, não se pode obrigar ao preenchimento de pelo menos um valor das categorias acima. O campo "T0" (que significa ausência de tumor maligno) foi excluído a pedido dos médicos que especificaram o documento, pois num Cancer Center, a totalidade dos pacientes submetidos à radioterapia serão oncológicos. As subcategorias do TNM (como T1a, T1mic) foram também excluídas por concisão.

Seguimento Oncoclínica: O TNM é apresentado em três campos de preenchimento livre, um para o estado T, um para o N e outro para M. Essa escolha foi feita pelos médicos que especificaram o documento, pois seria mais flexível e rápido que escolher numa longa lista de opções. A única restrição feita é o máximo de 5 caracteres. Foi feita uma contagem de respostas mais frequentes e 218 valores diferentes foram encontrados para esses campos. Focando nos valores mais frequentes (consideramos os valores que representam 90% dos documentos preenchidos), verificamos que a presença do

caractere “1” ou “l” no campo “Estadio T” indica para todos os casos levantados um caso de “T1”. Esses mapeamentos foram feitos para as categorias básicas (como na ficha de admissão da radioterapia).

Oncologia cutânea: para cada parâmetro T,N e M, um combo (um campo com grupo de respostas fechadas) com as opções existentes é apresentado, com as opções:

- TX, T0, T1, T2, T3, T4
- NX, N0, N1, N2, N3
- MX, M0, M1

Essa é a situação ideal, pois apenas uma opção pode ser escolhida por parâmetro, evitando inconsistências. Apesar desse tipo de campo poder ser marcado como obrigatório, decidiu-se o contrário, pois as informações necessárias para o estadiamento podem não estar disponíveis no momento do preenchimento do documento. Isso ocasionou um grande número de campos em branco.

Evolução ambulatorial: o TNM e estadio clínico, quando existente, é preenchido ao meio de um campo de texto livre, junto do estado atual, hipótese diagnóstica e tratamento. Para extrair essa informação dessas fichas, seriam necessárias técnicas avançadas de NLP, e como essa informação está disponível em outros lugares de maneira estruturada, não nos preocupamos em recuperá-la daqui.

Os documentos acima (exceto o de evolução ambulatorial) contém informação sobre o estadio TNM de mais de 3000 pacientes, atendidos desde agosto de 2009 (quando a primeira dessas fichas, a da Oncoclínica, entrou em operação), porém pelo visto acima, estão em formatação diferente, exigindo a aplicação de integração de dados. Apesar de os dados estarem guardados na

mesma estrutura de banco de dados e tabelas, sua representação diferente exige a aplicação dessas técnicas.

5.6.4 Axiomas de inferência de estadiamento

Usando a 6ª edição do TNM, implementamos as regras para inferência do estadiamento clínico baseado nos valores de T,N e M de pacientes de câncer de mama. As regras de estadiamento diferem dependendo do tipo de câncer. Portanto, criamos classes denominadas C50_ECIIIb, C50_ECIIIc e C50_ECIV para deixar claro que trata-se do estadiamento específico para câncer de mama.

Para a classe ECIV de câncer de mama, o TNM exige que a variável M tenha valor 1, indicando que há metástase à distância, não importanto o tamanho da lesão principa (T0-T4) ou o comprometimento linfonodal (N0-N3). Portanto, a regra de inferência ficou simplesmente: $C50 \sqcap M1 \sqsubseteq C50\_ECIV$

Para ECIIIb, é necessário T4 (tumor com invasão da parede torácica, pele, ou ambos), M0 (ausência de metástase) e comprometimento linfonodal N0, N1 ou N2 (ausência de comprometimento linfonodal, linfonodos das axilas contém células tumorais mas não estão acoplados a outros tecidos, ou tanto os linfonodos das axilas contém células tumorais e estão acoplados a outros tecidos quanto esses estão livres, porém há células cancerosas nos linfonodos abaixo do esterno), o que gerou as regras:

$$C50 \sqcap T4 \sqcap N0 \sqcap M0 \sqsubseteq C50\_ECIIIb$$

$$C50 \sqcap T4 \sqcap N1 \sqcap M0 \sqsubseteq C50\_ECIIIb$$

$$C50 \sqcap T4 \sqcap N2 \sqcap M0 \sqsubseteq C50\_ECIIIb$$

Para ECIIIc, é necessário que o câncer tenha classificação M0, N3

(comprometimento dos linfonodos mais distantes, como os claviculares, sobre o esterno ou ambos axilares e abaixo do esterno) e qualquer T. Assim, a regra ficou:

$$C50 \sqcap N3 \sqcap M0 \sqsubseteq C50\_ECIIIc$$

Esse tipo de regras permite a inferência em apenas um sentido; assim, se classificarmos a instância como C50_ECIV, não será possível concluir que essa instância pertence às classes C50 e M1. Para isso, seria necessário estabelecer um axioma de equivalência. No nosso caso específico, os dados brutos estão representados como TNM e precisamos realizar inferência apenas para, a partir dessas variáveis, concluir o estadiamento, então não representamos o axioma dessa maneira.

5.6.5 Mapeamento de tratamento quimioterápico

Os tratamentos quimioterápicos são prescritos de maneira estruturada através do sistema EHR pelos médicos do departamento de Oncologia Clínica, exceto para alguns medicamentos que seguem fluxo diferenciado, e para pacientes internados. Nessas exceções, a prescrição é feita em um documento em texto livre, sem codificação de tipo algum. Além disso, a prescrição sozinha não garante que o paciente tenha usado a medicação; para isso, seria necessário checar o controle de infusão, feito pelas enfermeiras da Oncologia Clínica, que não era feito em sistema. Decidimos considerar que toda prescrição é infundida, mesmo podendo incorrer em falsos positivos.

Como comentado em 5.5, todos os medicamentos que são prescritos no módulo de prescrição do sistema EHR são cadastrados em uma tabela específica, com nome comercial ou do princípio, apresentação e outras informações. Procuramos os nomes relacionados a medicamentos da classe dos taxanos utilizando a ontologia de medicamentos disponível em NCBO, e procuramos manualmente por esses nomes na tabela de ítens de prescrição.

Além do sistema EHR, há também o sistema RHC que contém informação sobre tratamento Quimioterápico. Porém, não há detalhamento sobre tipo de droga, dose ou duração do tratamento, apenas se o paciente foi tratado ou não com esse tipo de terapia. Decidimos incluir essa informação em nossa integração pois, apesar de não ter a mesma precisão que o banco EHR, pode-se criar um critério “relaxado”, em que conste Quimioterapia e não Taxanos, e ampliar a lista de pacientes potencialmente selecionáveis, apesar de aumentar um pouco o trabalho da equipe do estudo.

5.7 AMBIENTE EXPERIMENTAL

Nosso experimento consistiu em montar três sistemas de integração, um deles o sistema *Ontocloud* que está sendo proposto nessa tese, e outros dois que implementam técnicas usuais de integração de bancos de dados para investigar aspectos específicos do nosso sistema.

Para avaliar empiricamente a correção dos resultados de consultas obtidos pela inferência implementada pelo nosso sistema, compararemos os mesmos com os produzidos por uma *triplestore* com as mesmas capacidades de inferência que nosso sistema.

Para avaliar a velocidade com que *Ontocloud* produz os resultados, compararemos o tempo que este gastou para resolver consultas com o tempo utilizado por um sistema de integração de dados baseado em reescrita de consultas, porém utilizando bancos relacionais. Utilizaremos as mesmas consultas de mapeamento dos outros métodos, porém num banco de dados virtual que simula a inferência de conceitos apresentada. Também, comparando com este sistema, analisaremos os requisitos não funcionais de *Ontocloud*. Denominaremos este sistema *Federation*.

Replicamos os bancos de dados de origem utilizando o software Pentaho Data Integration Community Edition[1]; cada uma das tabelas relevantes em cada um dos bancos foi copiada para uma instância do banco de dados relacional PostgreSQL 8.3[2]. Uma vez que os bancos de origem são utilizados para atividades operacionais do A. C. Camargo Cancer Center, com novos dados sendo registrados o tempo todo e com consultas que por vezes desaceleram a velocidade de resposta dos bancos e tornam os resultados não comparáveis, optamos pela replicação, criando assim um ambiente controlado para o experimento.

Executamos uma série de consultas nos três sistemas, começando pela classe CriterioSelecao, que traz os pacientes que atendem ao critério de seleção, e CriterioSelecaoRelaxado, idêntico ao anterior porém especificando apenas Quimioterapia ao invés de Taxanos., e em seguida por algumas das classes que compõe este critério: M1, N0, T4, Quimioterapia, Taxanos, HER2positivo, C50 e AdenocarcinomaInvasivo. Executamos cada consulta em cada sistema 100 vezes, e registramos o tempo gasto e o número de resultados obtidos. Na primeira rodada, guardamos também a lista completa de resultados.

O servidor em que a configuração experimental foi implantada e os testes foram executados tinha 4 núcleos de processamento a 3.00GHz, 64 bits, e 8GB de RAM, rodando o sistema operacional CentOS 5. O servidor de banco de dados utilizado foi o PostgreSQL 8.3. Utilizamos também o Pentaho Data Integration Community Edition versão 4.0.1, ARQ 2.10.1[3], PHP 5.2.5[4], Java 1.6.0.23[5], Teiid 7.7[6], Teiid Designer 7.8[7], OntoP 1.8[8] e JBoss Application

[1] http://community.pentaho.com/
[2] http://www.postgresql.org/
[3] http://jena.apache.org/
[4] http://php.net/
[5] https://www.java.com/
[6] http://www.jboss.org/teiid/
[7] http://www.jboss.org/teiiddesigner
[8] http://ontop.inf.unibz.it/

Server 5.1.0 GA[9]. Durante os experimentos, o servidor não executou nenhum processo que não estivesse diretamente relacionado à execução dos testes e que pudesse interferir na aferição de tempo.

5.7.1 *Ontocloud*

Para mapear os objetos nos bancos de dados de origem nos conceitos da ontologia, utilizamos o software OntoP. Criamos arquivos de configuração descrevendo os pacientes como instâncias dos conceitos escrevendo arquivos de mapeamento entre ambos (utilizamos as consultas estabelecidas no Anexo A. Para cada um dos três bancos de origem configuramos uma fonte de dados OntoP no servidor *open-rdf-workbench*[10] (incluído na instalação do OntoP) com os mapeamentos separados. Testamos as fontes de dados rodando consultas de teste em cada uma delas.

Para estimativa do custo e para otimizar a ordem de execução das consultas federadas, fizemos consultas *SPARQL* especificando cada uma das classes em cada *endpoint*.

Construímos a ontologia de federação, necessária para esse método, relacionando os conceitos da ontologia de domínio a instâncias da classe System por meio da propriedade systemImplementsConcepts. Cada instância representa uma fonte de dados e tem seu nome e endereço do endpoint vinculadas pelas propriedades hasName e hasSparqlEndpoint.

Para a execução de consultas, utilizamos a biblioteca SPARQLFederator, cuja descrição do desenvolvimento está na seção 4.6. Ela foi chamada em linha de comando Linux especificando-se como argumentos: a ontologia de domínio, a ontologia de federação e o nome da classe (conceito) a qual queremos encontrar instâncias representando pacientes.

[9] `http://www.jboss.org/jbossas/`
[10] `http://www.openrdf.org/news.jsp`

5.7.2 *Federation*

Para a construção de um banco de dados virtual seguindo o modelo relacional, utilizamos o software open source Teiid, e o plugin Teiid designer para elaboração do ambiente. Primeiro, adicionamos à configuração do servidor JBoss as informações para conexão aos bancos de dados de origem. Essas conexões são referenciadas pelo banco de dados virtual. Em seguida, elaboramos uma visão (view) para cada um dos conceitos que fazem parte de CriterioSeleção. Utilizamos as consultas de mapeamento, com adaptações para o dialeto SQL adotado pelo Teiid. Em especial, as consultas que mapeiam os conceitos relacionados a HER2+ tiveram as funções específicas de PostgreSQL para expressão regulares substituídas por funções regulares de manipulação de texto. Para os conceitos que são definidos em mais de um banco de dados, unificamos as consultas SQL por meio da cláusula UNION, que retorna um conjunto de resultados equivalente à união dos resultados de cada consulta individual. Para os conceitos que são definidos em termos de intersecção de outros conceitos, utilizamos a cláusula JOIN, que tem esse comportamento. Por fim, construimos a visão CriterioSeleção. Uma vez terminada a construção das visões, publicamos o resultado no servidor e passamos a executar os testes.

5.7.3 *Triplestore*

Extraimos e traduzimos os bancos de dados de origem em formato RDF através da ferramenta *materializeSesame*, distribuída juntamente com o OntoP. Utilizamos os mesmos arquivos de mapeamento utilizados pelo sistema *Ontocloud*. Foram necessário 6 minutos e 25 segundos para completar a extração, que gerou 4.161.960 triplas. Para realizar as buscas, configuramos o software Fuseki com um assembler customizado, especificando um datasource composto pela ontologia de domínio (já utilizada pelo *Ontocloud*), pelos arquivos extraídos na etapa anterior, e um modelo de ontologia baseado na

especificação OWL_MEM_MICRO_RULE_INF, o menor subconjunto disponível no Jena de reasoner OWL, que implementa as duas regras de inferência de que necessitamos (ver seção 4.4.2). Esse modelo ao ser inicializado carrega os dados, as regras e os resultados de inferência em memória. No nosso caso específico, a instância Fuseki ocupou no máximo 1.8Gb de RAM durante as consultas.

5.8 RESULTADOS

Para avaliação do custo de cada consulta no sistema Ontocloud, executamos diversas consultas, uma para cada classe, diretamente nos *endpoints* e capturamos o resultado e tempo que cada uma levou. Os dados estão representados na Tabela 2. O modelo da consulta é como abaixo (substituindo-se `CLASS` pelo URI da classe apropriada).

```
SELECT count(distinct ?pct)
WHERE {
    ?pct a CLASS
}
```

Nosso experimento consistiu em executar 100 rodadas de uma sequência de 10 consultas diferentes, totalizando 1000 execuções. As execuções foram feitas em um sistema de integração por vez. Todas as consultas em todas as rodadas foram executadas sem erros pelos 3 sistemas. Consultas equivalentes trouxeram resultados totalmente equivalentes entre sistemas. Em particular, consultas executadas nos sistemas *Triplestore* e *Ontocloud* retornaram os mesmos resultados.

Verificamos que no caso específico do *Triplestore*, as duas primeiras consultas apresentaram tempo de execução de 4m58s e 2m3.8s, e as demais (até a milésima), tempos sempre abaixo de um segundo (ver Figura 12), não importando a consulta avaliada. Na parte direita do mesmo gráfico, vemos que

da execução 50 até a 1000 os tempos oscilam entre aproximadamente 0,60s e 0,80s.

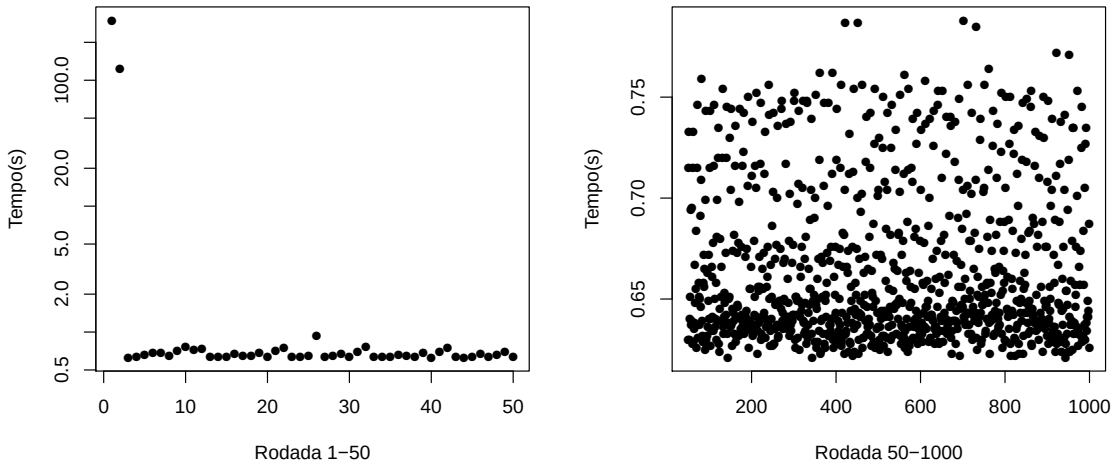

Figura 12 – Tempo de execução (em segundos) das 100 rodadas de 10 consultas executadas em *Triplestore*.

No Quadro 9, mostramos algumas classes relevantes para a classificação dos pacientes e marcamos se cada um dos 7 pacientes incluídos no estudo foi retornado pelo sistema de integração de dados ao se buscar pela classe. Na prática, os pacientes mostrados deveriam atender a todos os critérios, exceto os M1 e T4, o qual deve atender um dos dois.

Desses 7 pacientes, 5 foram encontrados pelos sistemas como pertencentes à classe CritérioTriagem. Um deles não atendeu ao critério Taxanos e outro não atendeu ao critério DoençaMetastática (M1) ou DoençaAvançada (T4). Analisando o prontuário desses pacientes, verificamos que o primeiro teve infusão de Taxano enquanto internado e por isso sua prescrição seguiu fluxo diferente dos demais, sendo descrita discursivamente em documento genérico. O segundo é um paciente com metástase, porém o campo M encontrava-se vazio; a indicação de metástase estava na descrição textual do caso.

No total, foram encontrados 96 pacientes únicos atendendo aos critérios de inclusão especificados, e 147 atendendo ao critério relaxado. Esses

pacientes poderiam representar a primeira listagem para busca manual, caso o projeto estivesse recrutando pacientes ainda.

Quadro 9 – Os 7 pacientes incluídos na pesquisa e conceitos de busca em que foram incluídos.

Classe	s1	s2	s3	s4	s5	s6	s7	Pacientes
CriterioTriagem	✓	✓	✓	X	✓	X	✓	96
CriterioRelaxado	✓	✓	✓	✓	✓	X	✓	147
HER2 positivo	✓	✓	✓	✓	✓	✓	✓	1.339
C50	✓	✓	✓	✓	✓	✓	✓	41.378
T4	✓	X	X	✓	X	X	X	3.362
N0	X	✓	X	X	✓	X	X	18.580
M1	X	✓	✓	✓	✓	X	✓	5.491
Adenocarcinoma Invasivo	✓	✓	✓	✓	✓	✓	✓	30.716
Taxanos	✓	✓	✓	X	✓	✓	✓	4.633
Quimioterapia	✓	✓	✓	✓	✓	✓	✓	13.591

Para determinar qual o método que resolveu determinada consulta com melhor desempenho, utilizamos dois critérios. Calculamos a razão entre a moda do tempo de execução do método *Ontocloud* e o tempo de execução do método *Federation*. Utilizamos a moda pois os dados não seguem a distribuição normal. Para verificar se a diferença entre as modas representa um desempenho consistentemente melhor, calculamos a Taxa de Erro de Bayes (BER), ou a razão da área da intersecção entre as curvas de densidade de probabilidade sobre a área da união das mesmas curvas. As funções de densidade de probabilidade foram ajustadas pelo método de kernel (Scott 1992). Finalmente, consideramos o nível de confiança de uma determinada comparação entre sistemas alto quando o BER calculado foi inferior a 0,15. Essas estatísticas estão descritas na Tabela 1.

Podemos ver na Figura 13 uma ilustração dessa comparação. À esquerda, razão entre a moda do tempo de execução da consulta no método *Ontocloud* e *Federation* (em azul, as consultas em que o método *Federation* foi mais veloz, e em vermelho, as consultas mais rápidas pelo *Ontocloud*). Para as consultas C50, N0, T4 e M1, apresentamos o gráfico da densidade

de distribuição dos tempos de execução do *Federation* (em azul) e *Ontocloud* (em vermelho).

Tabela 1 – Estatísticas da comparação entre tempos de execução de *Ontocloud* e *Federation*.

Classe	$Mo(Fed)$	$Mo(Ont)$	$\frac{Mo(Fed)}{Mo(Ont)}$	BER	Mais veloz	Confiança
CriterioSelecao	$102,42$	$4432,40$	$0,02$	$3,03 \times 10^{-17}$	*Federation*	Alta
CriterioSelecaoRelaxado	$105,00$	$4711,33$	$0,02$	$5,66 \times 10^{-17}$	*Federation*	Alta
HER2positivo	$29,99$	$44,15$	$0,68$	$3,25 \times 10^{-17}$	*Federation*	Alta
C50	$3,40$	$3,23$	$1,06$	$0,47$	*Ontocloud*	Baixa
T4	$2,96$	$2,79$	$1,06$	$0,049$	*Ontocloud*	Alta
N0	$3,95$	$3,66$	$1,08$	$0,138$	*Ontocloud*	Alta
M1	$3,48$	$3,08$	$1,13$	$0,005$	*Ontocloud*	Alta
AdenocarcinomaInvasivo	$98,36$	$53,77$	$1,83$	$5,45 \times 10^{-17}$	*Ontocloud*	Alta
Taxanos	$16,65$	$3,03$	$5,50$	$0,003$	*Ontocloud*	Alta
Quimioterapia	$59,86$	$4,86$	$12,31$	$0,006$	*Ontocloud*	Alta

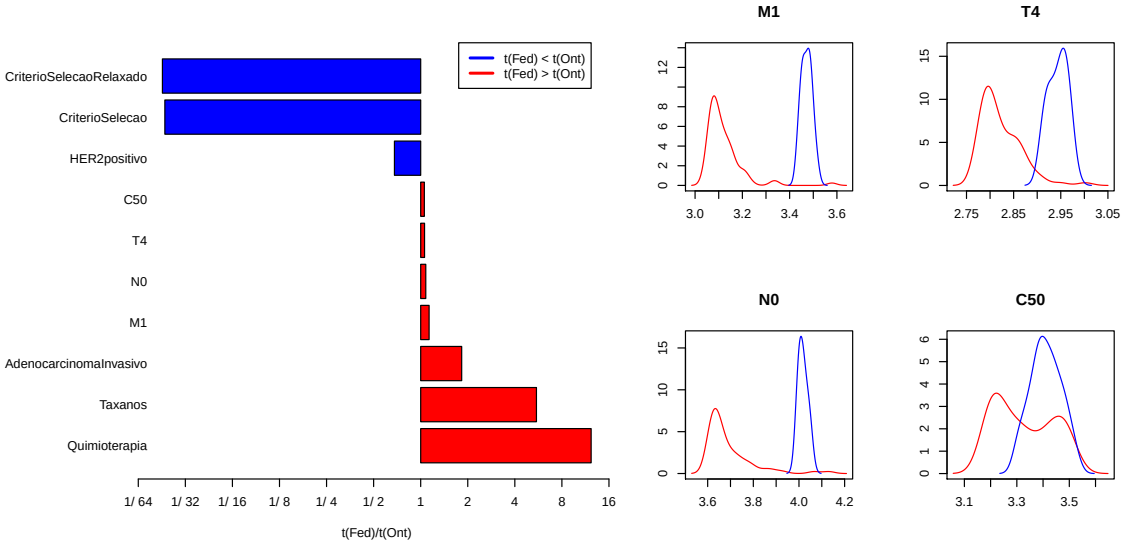

Figura 13 – Comparação de tempo para resolução de consultas por *Ontocloud* e *Federation*

Tabela 2 – Avaliação empírica de custo para planejamento de consultas no *Ontocloud*.

Endpoint	Classe	Resultados	Tempo(s)	Custo
AP	AdenocarcinomaInvasivo	380	6,25	0,59
AP	IHQ_HER2_Escore0	973	3,888	0,83
AP	IHQ_HER2_Escore1	551	3,864	0,77
AP	IHQ_HER2_Escore2	104	3,786	0,71
AP	IHQ_HER2_Escore3	270	3,817	0,73
AP	ISHRazaoHerCHR17Maior2,2	930	7,713	1,52
EHR	C50	15.076	545,516	101,75
EHR	Cetuximab	614	68,074	12,55
EHR	Docetaxel	1.941	80,807	15,04
EHR	Erlotinib	136	15,399	2,84
EHR	Gefitinib	5	61,177	11,22
EHR	M0	8.280	42,072	8,67
EHR	M1	2.757	18,184	3,65
EHR	N0	6.287	202,893	37,92
EHR	N1	3.109	8,42	1,90
EHR	N2	1.788	5,685	1,25
EHR	N3	710	3,882	0,79
EHR	Paciente	862.400	20,279	103,72
EHR	PacienteMaiorDe18Anos	415.419	9,569	49,92
EHR	Paclitaxel	2.940	140,997	26,19
EHR	T0	182	270,499	49,61
EHR	T1	3.671	167,47	31,13
EHR	T2	3.377	66,06	12,50
EHR	T3	3.018	46,995	8,96
EHR	T4	1.812	27,675	5,28
EHR	TaxanoHaMaisDeQuatroSemanas	2764	2,499	0,78
EHR	Trastuzumab	631	51,664	9,54
RHC	AdenocarcinomaInvasivo	9.280	1,366	1,33
RHC	C50	6.366	1,281	0,97
RHC	M0	20.505	1,446	2,64
RHC	M1	3.691	1,174	0,64
RHC	N0	16.449	1,449	2,17
RHC	N1	2.895	1,162	0,55
RHC	N2	1.771	1,091	0,41
RHC	N3	500	1,009	0,24
RHC	PacienteMaiorDe18Anos	38.050	2,37	4,85
RHC	Quimioterapia	10.850	1,352	1,51
RHC	T0	10	0,926	0,17
RHC	T1	8.266	1,308	1,20
RHC	T2	5.620	1,259	0,88
RHC	T3	3.765	1,229	0,66
RHC	T4	2.353	1,161	0,49

6 DISCUSSÃO

Ontocloud, por meio da combinação de expansão de consultas e camada OBDA, permite a implementação de inferência em federações de bancos relacionais. A etapa de expansão de consultas traduz os axiomas OWL em SPARQL, que por sua vez são traduzidos em SQL na etapa de OBDA. Assim, é possível acoplar esse *framework* de representação de conhecimento a bancos relacionais, ampliando as possibilidades de recuperação de informação. Apesar de ser possível realizar as mesmas operações em banco relacionais por meio de linguagem procedural, OWL permite uma representação formal declarativa, mais próxima da linguagem natural, o que facilita a construção e interpretação dos axiomas por especialistas da área médica.

Apesar de ser um protótipo, *Ontocloud* pode ser aplicado, na versão atual, a outros problemas de integração para projetos de pesquisa diferentes, ou mesmo problemas não relacionados a medicina. Sua limitação atual reside no tipo de axiomas que são expandidos, e na falta de suporte a propriedades além de `rdfs:type`. O sistema está publicado com licença de código aberto, podendo ser ampliado conforme a necessidade.

É importante distinguir a função do sistema de integração de dados de seus mapeamentos. A capacidade de qualquer sistema de integração de bancos de dados de responder as consultas feitas com alta especificidade e sensibilidade é diretamente dependente da qualidade da informação em suas origens e dos mapeamentos apresentados. Em um banco de dados utilizado operacionalmente em um grande hospital, existem diversos eventos e artefatos que comprometem a qualidade da informação: os documentos eletrônicos sofrem alterações ao longo dos anos de uso, profissionais com diferentes níveis de habilidade e treinamento em registro, ou mesmo eventos emergenciais

como queda de sistema ou energia.

Dos 7 pacientes incluídos de fato no estudo, 5 foram encontrados pelo *Ontocloud*. Dois pacientes não foram encontrados pois alguns dos dados necessários para definição do critério de seleção foram cadastrados de forma discursiva ao invés de se utilizar os campos estruturados que foram mapeados. Isto mostra a importância da utilização de métodos de NLP adequados na busca por pacientes, mesmo existindo dados estruturados.

Além dos 5 pacientes que foram incluídos de fato, outros 91 satisfazem os critérios de seleção. Todos esses pacientes poderiam ser avaliados manualmente pela equipe do estudo, ampliando as chances de recrutar outros participantes de pesquisa. De acordo com Fink et al. (2004), o número de pacientes recrutados utilizando-se ferramentas automatizadas pode ser até 250% maior em comparação com o método manual. Apesar de não podermos verificar diretamente essa afirmação, é razoável supor que, dos 91 pacientes encontrados pelo sistema, 17 atenderiam a todos os critérios e poderiam ser incluídos.

Criamos o conceito CriterioRelaxado ao verificarmos que nem todos os pacientes possuíam a droga específica que foi utilizada em seu tratamento cadastrada de forma estruturada. Muitos possuíam apenas a informação se foi feita quimioterapia ou não. De uma forma geral, os dados presentes nos bancos de dados considerados não representam toda a informação sobre o paciente. Portanto, quando um dos critérios não é atendido pela inspeção no banco de dados, não é impossível que o paciente atenda ao critério mas o dado simplesmente não esteja presente na forma esperada. Assim, a criação de critérios que utilizem categorias mais amplas de seleção podem prover uma lista de pacientes secundária, que certamente trará mais falsos positivos mas ampliará a cobertura.

O fato de replicarmos parcialmente os três bancos de dados em apenas

uma instância do PostgreSQL não alterou de maneira significativa os resultados em relação a estabelecer três instâncias, pois o mesmo é capaz de responder consultas simultâneas em tabelas diferentes. Ainda, os mapeamentos farão referências a três usuários diferentes dentro dos bancos, que só acessarão um conjunto de tabelas referentes a um sistema original, reproduzindo a situação real de segregação dos dados. Apesar de termos configurado todas as fontes de dados em apenas um servidor de aplicações JBoss, não consideramos que haveria diferença em comparação com a configuração de três servidores de aplicação, uma vez que *Ontocloud* não paraleliza consultas, enviando uma por vez para o servidor.

A equivalência dos resultados de consultas entre os sistemas *Ontocloud* e Triplestore validou empiricamente o algoritmo de inferência por expansão de consultas federadas, peça crucial do sistema *Ontocloud*. Seis consultas foram consistentemente mais rápidas no *Ontocloud* que no *Federation*, três foram consistentemente mais rápidas no *Federation* que no *Ontocloud* e uma (C50) teve desempenho melhor no *Ontocloud*, porém não de forma consistente.

Os três casos em que o sistema *Ontocloud* teve performance pior que *Federation* foram as consultas HER2positivo, CriterioTriagem e CriterioTriagemRelaxado. Na primeira, apenas um banco foi consultado pois nenhum outro continha mapeamentos para esse critério, portanto os mecanismos de integração adicionaram um custo adicional que foi melhor gerenciado pelo software de integração utilizado pelo *Federation*, o Teiid. Além disso, houve diferença nas implementações, já que o Teiid não possui funções de tratamento de expressões regulares tão avançadas como o PostgreSQL. Essa diferença provavelmente ocasionou uma consulta mais eficiente através do *Federation*.

A consulta CriterioTriagem e CriterioTriagemRelaxado, que envolve um grande número de outros conceitos, foi resolvida em pouco mais de um minuto para *Federation* e aproximadamente uma hora e vinte minutos para *Ontocloud*.

Isso se deve ao fato de que o Teiid tem um otimizador e executor de consultas especializado em bancos virtuais muito refinado, ao passo que *Ontocloud* faz poucas otimizações e não tem um executor de consultas adequado para federações. Por exemplo, o software FedX tem um algoritmo de intersecção(*JOIN*) adequado a federações, executando as duas consultas em paralelo e depois calculando a intersecção entre os resultados. Como *Ontocloud* utiliza o executor de consultas padrão do Jena, que pressupõe que os dados consultados tem todos o mesmo custo de recuperação, o desempenho dessas consultas, que são as mais complexas de todas, ficou muito aquém do desempenho pelo sistema *Federation*. Mas para o propósito definido para o sistema, que é o de selecionar possíveis participantes de pesquisa para um estudo, o tempo de execução de *Ontocloud* é aceitável, uma vez que retornará resultados baseados nos dados mais recentes possível.

Obtivemos sucesso na harmonização semântica dos conceitos relacionados a quimioterapia, por meio de axiomas que vinculam as classes mais específicas como subclasses das menos específicas. Assim, a classe relativa à droga Taxol é subclasse de Taxanos, que é subclasse de Quimioterapia. Os bancos de dados contém conceitos relativos ao nível mais específico ou ao mais geral, porém utilizando essa técnica, conseguimos que *Ontocloud*, ao buscar por Taxanos, traduzisse as consultas para as drogas específicas, assim como ao buscar por Quimioterapia, incluir os resultados do banco com a informação mais geral.

À exceção do mapeamento dos conceitos relacionados ao status HER2 do tumor, que por motivos técnicos implementaram em linguagem SQL os critérios patológicos de avaliação dessa variável, o conhecimento sobre câncer de mama necessário para a resolução do problema foi bem separado do mapeamento do banco de dados, preservando a semântica dos dados.

Comparando essa situação com a implementação dos mesmos conceitos em *Federation*, vemos que a consulta criada para Taxanos mistura as

drogas em uma única consulta. Assim, há perda de detalhes da informação, que poderia ser utilizada para conferência ou outras consultas mais específicas, além de tornar mais difícil e propenso a erros o processo de alterar a consulta, por exemplo, para adicionar ou remover uma droga dessa classificação. Para o conceito Quimioterapia, houve duplicação do código SQL escrito na consulta para taxanos, com a adição de outras drogas e de outro banco de dados. Essa duplicação de código resulta em dificuldade aumentada de manutenção no futuro, pois no caso de uma nova droga da classe dos taxanos ser adicionada, as duas consultas deverão ser atualizadas.

Poderíamos ter adotado uma abordagem hierárquica para a construção das consultas, criando inicialmente uma visão para cada droga, e a visão Taxanos que consolidaria os resultados das visões abaixo e assim por diante. Isso mitigaria a questão de manutenção das visões, embora não resolva o problema intrínseco da integração baseada em bancos relacionais, que é o de codificar conhecimento sobre o domínio em linguagem computacional.

Alguns dos trabalhos analisados aproveitaram-se de iniciativas de integração de bancos de dados (Kamal et al. 2005; Köpcke et al. 2013a) ou ferramenta de busca (Dugas et al. 2010) já existentes em suas instituições para a identificação de pacientes prospectivos. Isso limita a aplicabilidade dessas soluções, pois é necessário que já exista um *data warehouse* pronto para a consulta. Nesse ponto, o escopo de Ontocloud é mais amplo, pois pressupõe o uso dos bancos de dados utilizados em produção (ou uma réplica) para sua implantação.

Também, alguns utilizaram-se do expediente de enviar mensagens automaticamente aos pesquisadores responsáveis, quando um potencial participante de pesquisa é internado (Kamal et al. 2005) ou dá entrada pela emergência (Weiner et al. 2003). Uma funcionalidade semelhante pode ser implementada utilizando-se um agendador automático de tarefas que faça consultas periódicas em Ontocloud e envie os resultados por e-mail ou SMS.

No estudo Dugas et al. (2010) mencionou-se especificamente a necessidade de estudos de câncer utilizarem o tipo histológico e marcadores moleculares para uma adequada avaliação do paciente. No estudo Kamal et al. (2005), além da ferramenta apresentada ser voltada para estudos em câncer de mama, menciona-se que no *data warehouse* institucional são integrados textos discursivos de laudos de anatomia patológica. No nosso caso do sistema, incluímos tipos histológicos específicos e os resultados de um marcador molecular medido por dois exames diferentes, a partir de textos discursivos dos laudos de anatomia patológica. Para isso, utilizamos técnicas básicas de busca e análise de texto. Essa experiência pode ser utilizada como base para inclusão de métodos de Processamento de Linguagem Natural mais avançados, de forma a extrair maiores detalhes de narrativas médicas e adicioná-las ao sistema de integração.

7 CONCLUSÃO

Apresentamos um novo sistema, *Ontocloud*, para integração de dados, modelado a partir das necessidades verificadas no problema de selecionar participantes de pesquisa a partir da população de um hospital. *Ontocloud* permite a realização de inferência em federações de bancos de dados relacionais por meio de reescrita de consultas e mapeamento entre ontologia e banco relacional. Essas características permitem que *Ontocloud* traga resultados sempre atualizados, consolide diferentes bancos de dados, harmonize suas representações e permita a inferência de novas informações. A performance de *Ontocloud* mostrou-se satisfatória, considerando-se que é um sistema protótipo. Devotando-se esforços em otimização e planejamento, seu desempenho poderá equiparar-se ou mesmo superar outros sistemas de integração de dados.

Como trabalho futuro, o sistema pode ser expandido para implementar todo o perfil OWL-QL e EL, aumentando a gama de aplicações que poderá atender. Quanto à performance, pode-se utilizar algumas das estratégias específicas para consultas federadas, como execução concorrente de cláusulas, implementar algoritmos de planejamento, e ampliar o rol de otimizações desempenhadas.

8 REFERÊNCIAS BIBLIOGRÁFICAS

Batini C, Lenzerini M e Navathe SB, A comparative analysis of methodologies for database schema integration. ACM computing surveys (CSUR) 1986; 18(4):323–364.

Beale T, The Health Record - why is it so hard? In: R Haux e C Kulikowski, editores, IMIA Yearbook of Medical Informatics 2005, Stuttgart: Schattauer, p. 301–304.

Beck AR, Cohn AG, Sanderson M, Ramage S, Tagg C, Fu G, Bennett B e Stell JG, Universities of Leeds , Sheffield and York UK utility data integration : overcoming schematic heterogeneity. 71431, p. 71431Z–71431Z–11.

Berners-Lee T, Connolly D, Kagal L, Scharf Y e Hendler J, N3Logic: A logical framework for the World Wide Web. Theory and Practice of Logic Programming 2008; 8(03).

Bizer C e Seaborne A, D2RQ-treating non-RDF databases as virtual RDF graphs. In: Proceedings of the 3rd International Semantic Web Conference (ISWC2004).

Brachman RJ e Levesque HJ, Knowledge representation and reasoning. Morgan Kaufmann 2004.

Buccella A, Cechich A e Brisaboa N, Ontology-Based Data Integration Methods: A Framework for Comparison. Colombian Journal of Computation 2005; 6(2):62–68.

Calvanese D, Giacomo GD, Lembo D e Lenzerini M, MASTRO - I: Efficient integration of relational data through DL ontologies. In: Description Logics, p. 227–234.

Casanova MA e Vidal VMP, Towards a sound view integration methodology. In: Proceedings of the 2nd ACM SIGACT-SIGMOD symposium on Principles of database systems - PODS '83, New York, New York, USA: ACM Press, p. 36.

Chen PPS, The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems 1976; 1(1):9–36.

Codd EF, A relational model of data for large shared data banks. MD computing: computers in medical practice 1970; 15(3):162–6.

Cormen T, Leiserson C, Rivest R e Stein C, Introduction to algorithms. The MIT press, 3rd edição 2001.

Cruz IF e Xiao H, The role of ontologies in data integration. Journal of engineering inteligent systems 2005; 13(4):854–863.

Cure O e Bensaid JD, Integration of relational databases into OWL knowledge bases: demonstration of the DBOM system. In: 2008 IEEE 24th International Conference on Data Engineering Workshop, IEEE, p. 230–233.

de Lusignan S, Codes, classifications, terminologies and nomenclatures: definition, development and application in practice. Informatics in primary care 2005; 13(1):65–70.

Dugas M, Lange M, Müller-Tidow C, Kirchhof P e Prokosch HU, Routine data from hospital information systems can support patient recruitment for clinical studies. Clinical trials (London, England) 2010; 7(2):183–9.

Fink E, Kokku PK, Nikiforou S, Hall LO, Goldgof DB e Krischer JP, Selection of patients for clinical trials: an interactive web-based system. Artificial intelligence in medicine 2004; 31(3):241–54.

Fitting M, First-Order Logic and Automated Theorem Proving. Springer, 2 edição 1995.

Fletcher B, Gheorghe A, Moore D, Wilson S e Damery S, Improving the recruitment activity of clinicians in randomised controlled trials: a systematic review. BMJ open 2012; 2(1):e000496.

Florescu A, Amir E, Bouganim N e Clemons M, Immune therapy for breast cancer in 2010-hype or hope? Current oncology (Toronto, Ont) 2011; 18(1):e9–e18.

GLICO - Grupo Latino Americano de Investigacoes Clinicas em Oncologia, Safety Study in Subjects With Metastatic Breast Cancer Who Progressed After Taxanes Treatment. (GLICO-0801) 2014.

Grob GN, Origins of DSM-I: a study in appearance and reality. The American journal of psychiatry 1991; 148(4):421–31.

Gruber TR, A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 1993; 5(April):199–220.

Haas LM, Lin ET e Roth Ma, Data integration through database federation. IBM Systems Journal 2002; 41(4):578–596.

Hamosh A, Scott AF, Amberger JS, Bocchini CA e McKusick VA, Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research 2005; 33(Database issue):D514–7.

Hudis CA, Trastuzumab–mechanism of action and use in clinical practice. The New England journal of medicine 2007; 357(1):39–51.

Hughes T, Bell G, Bloch E, Bressler R, David P, Denicoff M, Hounshell D, Joel AE, Lenois T, McIlroy D, Pugh E, Seitz C e Thacker C, Funding a Revolution: Government Support for Computing Research. The National Academies Press 1999.

Hull R, Managing semantic heterogeneity in databases. In: Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems - PODS '97, New York, New York, USA: ACM Press, p. 51–61.

Inmon WH, Evolution of decision support systems. In: Building the Data Warehouse, Wiley, 4th edição 2005; p. 576.

Junttila K, Meretoja R, Seppälä A, Tolppanen EM, Ala-Nikkola T e Silvennoinen L, Data warehouse approach to nursing management. Journal of nursing management 2007; 15(2):155–61.

Kamal J, Pasuparthi K, Rogers P, Buskirk J e Mekhjian H, Using an information warehouse to screen patients for clinical trials: a prototype. AMIA Annual Symposium proceedings / AMIA Symposium AMIA Symposium 2005; 5(6):1004.

Köpcke F, Kraus S, Scholler A, Nau C, Schüttler J, Prokosch HU e Ganslandt T, Secondary use of routinely collected patient data in a clinical trial: an evaluation of the effects on patient recruitment and data acquisition. International journal of medical informatics 2013a; 82(3):185–92.

Köpcke F, Trinczek B, Majeed RW, Schreiweis B, Wenk J, Leusch T, Ganslandt T, Ohmann C, Bergh B, Röhrig R, Dugas M e Prokosch HU, Evaluation of data completeness in the electronic health record for the purpose of patient recruitment into clinical trials: a retrospective analysis of element presence. BMC medical informatics and decision making 2013b; 13(37):37.

Kumar A e Smith B, Oncology ontology in the NCI thesaurus. Lecture notes in computer science 2005; 3581:213.

Lenzerini M, Data integration: a theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS'02, New York, New York, USA: ACM Press, p. 233–246.

McKusick VA, Mendelian Inheritance in Man. A Catalog of Human Genes and Genetic Disorders. Johns Hopkins University Press, Baltimore, MD, 12 edição 1998.

Min H, Manion FJ, Goralczyk E, Wong YN, Ross E e Beck JR, Integration of prostate cancer clinical data using an ontology. Journal of biomedical informatics 2009; 42(6):1035–45.

Mitri Z, Constantine T e O'Regan R, The HER2 Receptor in Breast Cancer: Pathophysiology, Clinical Use, and New Advances in Therapy. Chemotherapy research and practice 2012; 2012:743193.

Moja L, Tagliabue L, Balduzzi S, Parmelli E, Pistotti V, Guarneri V e D'Amico R, Trastuzumab containing regimens for early breast cancer. The Cochrane database of systematic reviews 2012; 4:CD006243.

Muranaga F, Kumamoto I e Uto Y, Development of hospital data warehouse for cost analysis of DPC based on medical costs. Methods of information in medicine 2007; 46(6):679–85.

Nahta R e Esteva FJ, HER-2-targeted therapy: lessons learned and future directions. Clinical cancer research : an official journal of the American Association for Cancer Research 2003; 9(14):5078–84.

Prokosch HU e Ganslandt T, Perspectives for Medical Informatics. Methods of Information in Medicine 2009; :38–44.

Pullokkaran LJ, in Business Intelligence Analysis of Data Virtualization Enterprise Data Standardization in Business Intelligence. Master thesis, Massachusetts Institute of Technology 2013.

Ramick DC, Data warehousing in disease management programs. Journal of healthcare information management : JHIM 2001; 15(2):99–105.

Robinson JA, A Machine-Oriented Logic Based on the Resolution Principle. Journal of the ACM 1965; 12(1):23–41.

Rubin DL e Desser TS, A data warehouse for integrating radiologic and pathologic data. Journal of the American College of Radiology : JACR 2008; 5(3):210–7.

Russell S e Norvig P, Artificial intelligence: a modern approach. Prentice hall Englewood Cliffs, NJ 1995.

Russell S e Norvig P, Artificial Intelligence: A Modern Approach. Pearson Education 2003.

Sahoo U e Bhatt A, Electronic data capture (EDC)–a new mantra for clinical trials. Quality assurance (San Diego, Calif) 2003; 10(3-4):117–21.

Sariego J, Breast cancer in the young patient. The American surgeon 2010; 76(12):1397–400.

Schmidt-SchaußM, Subsumption in KL-ONE is Undecidable. In: KR, p. 421–431.

Schmier JK, Kane DW e Halpern MT, Practical applications of usability theory to electronic data collection for clinical trials. Contemporary clinical trials 2005; 26(3):376–85.

Schober D, Boeker M, Schulz S e Tudose I, Developing DCO: The DebugIT core ontology for antibiotics resistence modelling. In: N Collier, U Hahn, D Rebholz-Schuhmann, F Rinaldi e S Pyysalo, editores, Proceedings of the Fourth International Symposium for Semantic Mining in Biomedicine, Cambridge, United Kingdom: CEUR-WS.org.

Schwarte A, Haase P e Hose K, FedX: a federation layer for distributed query processing on linked open data. ESWC 2011a; 2:481–486.

Schwarte A, Haase P e Hose K, FedX: optimization techniques for federated query processing on linked data. International Semantic Web Conference 2011b; 1:601–616.

Scott DW, Multivariate Density Estimation. Theory, Practice and Visualization. In: Multivariate Density Estimation. Theory, Practice and Visualization, John Wiley & Sons, capítulo 6, 1 edição 1992; p. 125–181.

Sheth AP e Larson JA, Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys 1990; 22(3):183–236.

Spackman KA, SNOMED CT milestones: endorsements are added to already-impressive standards credentials. Healthcare informatics : the business magazine for information and communication systems 2004; 21(9):54, 56.

Spackman KA, Campbell KE e Côté RA, SNOMED RT: a reference terminology for health care. Proceedings : a conference of the American Medical Informatics Association / AMIA Annual Fall Symposium AMIA Fall Symposium 1997; :640–4.

Sujansky W, Heterogeneous Database Integration in Biomedicine. Journal of Biomedical Informatics 2002; 34(2001):285–298.

Tan M e Yu D, Molecular mechanisms of erbB2-mediated breast cancer chemoresistance. Advances in experimental medicine and biology 2007; 608:119–29.

Teodoro D, Pasche E, Wipfli R, Gobeill J, Choquet R, Daniel C, Rucha P e Lovis C, Integration of biomedical data using federated databases. Swiss medical informatics 2009; 25(67):57–60.

ter Horst HJ, Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Web Semantics: Science, Services and Agents on the World Wide Web 2005; 3(2-3):79–115.

Velázquez I, Navarro X e Cobos A, Electronic data capture - Impact on the quality of the clinical research. Medicina clínica 2004; 122 Suppl:11–5.

Wache H, Voegele T, Visser U, Stuckenschmidt H, Schuster G, Neumann H, Hübner S e Vögele T, Ontology-based integration of information-a survey of existing approaches. In: Proceedings of IJCAI-01 Workshop on E-Business & the Intelligent Web, Seattle - WA: Citeseer, volume 2001, p. 108–117.

Weiner DL, Butte AJ, Hibberd PL e Fleisher GR, Computerized recruiting for clinical trials in real time. Annals of emergency medicine 2003; 41(2):242–6.

Welker JA, Implementation of electronic data capture systems: barriers and solutions. Contemporary clinical trials 2007; 28(3):329–36.

Yuhanna N e Gilpin M, The Forrester Wave™: Data Virtualization, Q1 2012. Relatório Técnico, Forrester 2012.

Zhang Q, Matsumura Y, Teratani T, Yoshimoto S, Mineno T, Nakagawa K, Nagahama M, Kuwata S e Takeda H, The application of an institutional clinical data warehouse to the assessment of adverse drug reactions (ADRs). Evaluation of aminoglycoside and cephalosporin associated nephrotoxicity. Methods of information in medicine 2007; 46(5):516–22.

Anexo 1 - Tópicos computacionais

REPRESENTAÇÃO FORMAL DE ONTOLOGIAS

Existe um grande número de formalismos para representar as relações em uma ontologia. Nesta tese, empregamos um baseado em lógicas de descrição, a qual explicamos sucintamente a seguir.

Os conceitos (classes, instâncias e propriedades) são escritos de maneira usual, porém as classes tem sua primeira letra em maiúsculas, e instâncias e propriedades iniciam em minúsculas. No caso de palavras compostas como nome de um único conceito, elas são escritas sem espaçamento, porém a próxima letra após a localização do espaço fica em maiúsculas.

- Classes: $Pessoa$, $Homem$, $Mulher$, $Musico$, $Guitarrista$, $Vocalista$, $Groupie$
- Instâncias: $johnLennon$, $yokoOno$
- Propriedades: $temNome$, $casadoCom$

A seguir exemplificamos axiomas que relacionam os conceitos descritos. Para indicar que uma instância pertence a uma classe, utilizamos a notação:

$Mulher(yokoOno)$

Para relacionar duas instâncias por meio de uma propriedade:

$casadoCom(johnLennon, yokoOno)$

Podemos afirmar que uma instância não pertence a determinada classe:

$\neg Mulher(johnLennon)$

O relacionamento de classe e subclasse é indicado como a seguir:

$Guitarrista \sqsubseteq Musico$

$Musico \sqsubseteq Pessoa$

A declaração a seguir indica que uma instância pertence simultaneamente a duas classes:

$Guitarrista \sqcap Vocalista(johnLennon)$

Para indicar o conceito cuja instâncias apenas se relacionam com instâncias de determinada classe ($Musico$) através da propriedade $casadoCom$:

$(\forall casadoCom.Musico) \sqsubseteq Groupie$

Neste último exemplo, $\forall casadoCom.Musico$ indica todas as instâncias que estão relacionadas com outra instância, pertencente à classe $Musico$ através da propriedade $casadoCom$. Assim, o axioma acima significa que todas as instâncias que possuem esse tipo de relação também pertencem à classe $Groupie$.

INFERÊNCIA

Inferência é o processo pelo qual nova informação é derivada de declarações e axiomas(Russell e Norvig 2003). No exemplo acima, a partir das afirmações $Guitarrista \sqcap Vocalista(johnLennon)$ e $Guitarrista \sqsubseteq Musico$, é possível concluir que $Musico(johnLennon)$. Após a avaliação desta nova afirmação, a mesma é adicionada à base de conhecimentos e pode ser utilizada para novas conclusões. Assim, $Musico(johnLennon)$ e $casadoCom(johnLennon, yokoOno)$, juntamente com o axioma $(\forall casadoCom.Musico) \sqsubseteq Groupie$, permite concluir que $Groupie(yokoOno)$. Atualmente, softwares para realização automática de inferência são baseados em dois algoritmos, o de *resolução*(Robinson 1965) e o de *tableaux* (Brachman e Levesque 2004; Fitting 1995).

COMPLEXIDADE DE ALGORITMOS

Neste ponto, é necessária uma pequena digressão para comentar um pouco sobre complexidade de algoritmos, tema que trata de estimar a quantidade de recursos (processamento ou memória) necessários para execução de um programa em função do tamanho de sua entrada (ou a quantidade de informação que é necessária para a execução do mesmo). A complexidade de tempo é representada na notação $O(f(x))$, onde $f(x)$ é uma função do tamanho da entrada do algoritmo. Assim, um algoritmo de complexidade $O(x^2)$ levará tempo proporcional ao quadrado da quantidade de informação recebida.

Classes de complexidade são grupos de funções de complexidade de algoritmos. A classe *P* denota problemas que tem complexidade da ordem de $O(x^k)$, onde k é alguma constante - diz-se que levam *tempo polinomial* para serem resolvidos ou são eficientes. Problemas da classe *NP* são aqueles em que a *verificação* de uma solução leva tempo polinomial, porém não necessariamente o cálculo da solução. Problemas da classe *co-NP* são aqueles em que a solução do *complemento* de um problema está na classe *NP* - sendo que, para uma entrada *A*, um problema de decisão χ que resulte numa resposta verdadeira, seu complemento $\bar{\chi}$ retornará uma resposta falsa, e vice versa.

A classe *NP* contém os problemas *P*. A classe *NP-difícil* contém os problemas tão ou mais difíceis quanto o mais difícil da classe *NP*. Por fim, *NP-Completo* são os problemas que pertencem às classes *NP* e *NP-difícil* simultaneamente. Atualmente, é desconhecido se $P = NP$ ou $P \neq NP$; caso sejam iguais, isso implicaria que os problemas *NP-Completo* poderiam ser resolvidos de forma eficiente(Cormen et al. 2001).

Podemos dar como exemplo de problema *NP-Completo* o do *caixeiro viajante*, em que um caixeiro viajante deverá visitar um certo conjunto de cidades, sem passar duas vezes pela mesma cidade. O problema *NP-Completo*

em questão é o de, dado um percurso de tamanho L, determinar um trajeto que resulte num percurso inferior. Uma possível solução para encontrar o percurso envolve listar todas as combinações possíveis de cidades ($n!$) e calcular o percurso para cada uma. Porém, verificar se o percurso de uma determinada solução é inferior a L é possível em tempo polinomial.

RDF: DESCRIÇÃO DE RECURSOS

RDF (Resource Description Framework) é um modelo para troca de dados na web[1]. RDF descreve *recursos*, relacionando-os com outros recursos ou com literais (como números ou sequencias de caracteres). Recursos são representados como URIs (Universal Resource Identifier), que são semelhantes em forma às URLs (Uniform Resource Locator) que identificam sítios da internet. As declarações de um documento RDF seguem o padrão:

```
<sujeito> <predicado> <objeto>
```

`Sujeito` e `predicado` devem ser URIs, enquanto que `objeto` pode ser um URI, um literal ou mesmo um nó vazio. Esta construção é chamada *tripla* e significa, caso `objeto` seja um recurso, que `sujeito` relaciona-se com `objeto` por meio da propriedade `predicado`. Se `objeto` for um literal, então `sujeito` tem uma propriedade `predicado` cujo valor é `objeto`. Em RDF, os literais são sempre sequencias de caracteres. Um documento RDF pode ser armazenado em bancos de dados conhecidos como *triplestores*[2], ou em arquivos, seguindo diferentes sintaxes. A sintaxe RDF/XML, descrita na documentação oficial[3], utiliza-se do padrão XML[4], mas também é popular o formato Notation3 (N3)(Berners-Lee et al. 2008), que facilita a leitura e escrita dos dados por pessoas.

[1] http://www.w3.org/RDF/
[2] literalmente, “armazéns de triplas”.
[3] `http://www.w3.org/TR/rdf-primer/`
[4] `http://www.w3.org/TR/REC-xml/`

RDFS, OWL: ONTOLOGIAS E INFERÊNCIA

RDFS (Resource Description Framework Schema) adiciona uma camada de semântica ao RDF, definindo classes, propriedades e como determinados recursos devem ser interpretados. Por exemplo, uma classe A ser subclasse de outra B implicar que as instâncias de A também são instâncias de B está definido na regra *rdfs9* do documento `http://www.w3.org/TR/rdf-mt/`. Estas descrições de interpretações (ou *semântica*) são implementadas por softwares conhecidos como *motores de inferência* que tornam explícito o conhecimento implícito em uma base de conhecimento. A inferência de RDFs é decidível, *NP-completo*, e pode ser resolvido mesmo em tempo polinomial (*P*) caso não haja nós vazios(ter Horst 2005).

OWL 2 (Web Ontology Language) tem suas fundações no RDFS e amplia sua semântica, de forma a poder representar conhecimento complexo e rico sobre coisas, grupos de coisas e suas relações. É baseado em lógicas de descrição, e construída de forma a permitir o uso do conhecimento por programas de computador para checagem de consistência ou realização de inferência [5].

Abaixo são relacionados alguns tipos de axiomas que podem ser representados em OWL 2. Note que é possível criar combinações entre os axiomas.

- *Intersecção*: Uma instância que pertença a duas classes A e B, pertence à interseção entre ambas.

- *Disjunção*: A disjunção entre duas classes A e B significa que não existe nenhuma instância que pertença às duas simultaneamente.

[5] `http://www.w3.org/TR/2012/REC-owl2-primer-20121211/`

- *Complemento*: Uma instância que pertença ao complemento de uma classe A *não pertence* a esta classe - podemos interpretar isso como *negação*.

- *Quantificação universal*: Este tipo de restrição categoriza instâncias que relacionem-se com outras instâncias, sempre pertencentes à mesma classe. No exemplo dado, a definição da classe $Groupie$ utiliza-se deste recurso.

- *Propriedades transitivas*: Uma propriedade p ser transitiva significa que se $p(a, b)$ e $p(b, c)$, então $p(a, c)$ é verdadeiro. Por exemplo, a relação $<$ entre números reais ou $temAntepassado$, em relações familiares, são propriedades transitivas.

- *Propriedades encadeadas*: Este axioma é uma espécie de generalização da transitividade de propriedades. Um axioma $p \circ q \rightarrow r$ significa que, se $p(a, b)$ e $q(b, c)$, então $r(a, c)$. Como exemplo, podemos dizer que $temPai \circ temIrmao \rightarrow temTio$.

Além dos axiomas que OWL 2 provê, existem também *regras*, que permitem a expressão de axiomas do tipo $a_1, ..., a_n \rightarrow b_1, ..., b_m$, onde a_i e b_j são *átomos* da forma $C(x)$, $p(x, y)$, com propriedades especiais $sameAs$ e $differentFrom$, que distinguem ou diferenciam entidades. Também podem ser utilizadas propriedades que realizam aritmética de datas, análise de sequencia de caracteres ou cálculos. As regras permitem mais expressividade que os axiomas de OWL 2, porém caso não haja restrições, torna-se indecidível (Schmidt-Schauß 1989)[6].

O uso de todas as capacidades que a especificação OWL 2 oferece pode tornar a computação destes problemas muito custosa, ou mesmo *indecidível*, termo utilizado para descrever a situação em que é impossível provar

[6] http://www.w3.org/Submission/SWRL/

que um problema tem ou não uma solução. Mesmo para problemas que sejam decidíveis, sua resolução pode levar tanto tempo ou utilizar tanta memória do computador que tornam-se impraticáveis. Por isso, existem diversas subdivisões da OWL 2, conforme vemos abaixo.

- OWL 2 DL: é o maior subconjunto *decidível* de OWL 2, de forma a tornar possível o uso de motores de inferência.
- OWL 2 EL: foi criada com o objetivo específico de representar grandes ontologias de saúde, tais como SNOMED-CT e NCI Thesaurus. É um subconjunto de OWL 2 DL, removendo-se a capacidade representar negações, disjunções, propriedades inversas e quantificação universal em propriedades.
- OWL 2 QL: foi elaborada tendo em mente seu uso acoplado com bancos de dados relacionais, em específico, para mapeamento de tabelas em ontologias e conversão de consultas, implementando inferência. Permite especificação de disjunção e e propriedades inversas, embora limite a construção de axiomas com intersecções de classes.
- OWL 2 RL: subconjunto da OWL 2 que é propícia para implementação utilizando-se de motores de inferência baseados em regras, de forma a prover inferência eficiente com perda mínima de expressividade.

SPARQL

SPARQL[7] é uma especificação de linguagem de consulta para RDF. É, assim como RDF e OWL, uma especificação mantida pela W3C e vem se tornando o padrão *de facto* para consultas em dados RDF. Em seguida, detalharemos as consultas do tipo SELECT, que recuperam informações (existem

[7] `http://www.w3.org/TR/rdf-sparql-query/`

também consultas ASK, que respondem se é possível responder uma consulta ou não, DESCRIBE, que detalha um determinado recurso, CONSTRUCT, que gera novas triplas). *Triplestores*, para responder a consultas SPARQL, disponibilizam um serviço conhecido como *endpoint SPARQL*, que provê respostas a consultas nesta linguagem.

Da mesma forma que a cláusula homônima no padrão SQL, estas consultas tem o padrão abaixo:

```
PREFIX prefixo: <URI-BASE>

SELECT ?campo1 ... ?campon
WHERE
{
   <sujeito> <predicado> <objeto>.
   ...
   FILTER( condicoes )
}
```

No início da consulta, pode-se especificar *prefixos*, que provem um modo prático de especificar URIs extensas; por exemplo, `<http://www.w3.org/2000/01/rdf-schema#subClassOf>` pode ser escrito como rdfs:subClassOf, ao se especificar que o prefixo rdfs: corresponde a `<http://www.w3.org/2000/01/rdf-schema#>`. Após a cláusula SELECT, especifica-se a lista de variáveis, iniciadas pelo caracter '?', que são de interesse. Dentro da cláusula WHERE, *padrões de triplas* especificam quais as triplas que devem ser buscadas no RDF. No papel de sujeito, predicado ou objeto podem estar URIs ou variáveis. Uma variável pode aparecer em mais de um padrão de tripla, ou nas

condições do filtro, restringindo os resultados. No exemplo abaixo, consultamos o recurso dbpedia [8] para obter o nome e data de nascimento de todos os membros do grupo The Beatles que já atuaram em filmes.

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX p: <http://dbpedia.org/property/>
PREFIX c: <http://dbpedia.org/class/yago/>
select ?name ?dob
{
 ?x a c:TheBeatlesMembers.
 ?x a c:EnglishFilmActors.
 ?x p:name ?name.
 ?x p:birthDate ?dob.
}
```

A especificação SPARQL 1.1 inclui a cláusula SERVICE que permite acesso a diferentes *endpoints* simultaneamente, possibilitando trazer resultados combinados destas diferentes fontes. Se desejarmos, por exemplo, combinar a consulta anterior com outro repositório de informações, o linkedmdb[9], para obter informações sobre os filmes em que estes Beatles atuaram, poderíamos escrever:

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX p: <http://dbpedia.org/property/>
PREFIX c: <http://dbpedia.org/class/yago/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

select ?name ?dob ?mdb1 ?mdb2 ?mdb3 ?mdb4
{
```

[8] http://dbpedia.org/sparql
[9] http://data.linkedmdb.org/

```
 SERVICE <http://dbpedia.org/sparql> {
?beatle a c:TheBeatlesMembers.
?beatle a c:EnglishFilmActors.
?beatle p:name ?name.
?beatle p:birthDate ?dob.
 }

 SERVICE <http://data.linkedmdb.org/sparql> {
        ?actor owl:sameAs ?beatle.
        { ?mdb1 ?mdb2 ?actor.  } UNION
        { ?actor ?mdb3 ?mdb4. }
 }

}
```

Na consulta acima, utilizamos duas vezes a palavra-chave SERVICE, que especifica em que *endpoint* uma determinada subconsulta deve ser realizada. A variável ?beatle é especificada em ambas subconsultas, estabelecendo uma ligação entre os dois serviços, e trazendo portanto informações vinculadas. Na segunda subconsulta, existem três padrões de triplas; a primeira vincula a variável ?actor à variável ?beatle, especificada na primeira subconsulta, por meio de owl:sameAs, uma relação que indica que os conceitos são equivalentes. O segundo e terceiro padrões trazem triplas que especificam o ator desejado seja como objeto, seja como sujeito, trazendo assim todas as informações disponíveis. A cláusula UNION traz os resultados retornados por um ou outro padrão de tripla, trazendo a totalidade de informações em linkedmdb sobre as pessoas referenciadas em dbpedia.

PERSPECTIVA TEÓRICA DA INTEGRAÇÃO DE BANCOS DE DADOS

Segundo Lenzerini (2002), um problema de integração de bancos de dados pode ser descrito como

$$I = \langle G, S, M \rangle$$

onde:

- S é o esquema de origem, definido sobre uma linguagem L_S;
- G é o esquema global, consolidado, que representa a visão unificada que se deseja, expressa em uma linguagem L_G (por exemplo, relacional ou ontologia);
- M os mapeamentos entre S e G, representados por um conjunto de asserções da forma:

$$q_S \rightsquigarrow q_G$$

$$q_G \rightsquigarrow q_S$$

sendo que q_G é uma consulta expressa na linguagem $L_{M,G}$ sobre o esquema G e q_S expressa na linguagem $L_{M,S}$ sobre o esquema de origem S. Consultar I equivale a apresentar uma consulta q_G ao esquema global. Interpretamos que um mapeamento $q_S \rightsquigarrow q_G$ indica que o conceito representado no esquema de origem pela consulta q_S é correspondente ao conceito representado no esquema global pela consulta q_G.

Um banco de dados para um dado esquema E é um conjunto de coleções de conjuntos; na abordagem relacional, cada coleção corresponde a um

esquema e cada conjunto que este contém representa uma tupla. Consideremos um banco de dados D que está de acordo e atende a todas as restrições do esquema de origem S. Baseado em D, podemos especificar o conteúdo do esquema global G. Chamamos um banco de dados global para I qualquer banco de dados para G.

Dado um banco de dados de origem D para I, a resposta $q_{I,D}$ à query q em I com respeito a D é o conjunto de tuplas t que pertencem a q_B para todos os bancos de dados globais B que são legais em I com respeito a D. $q_{I,D}$ é chamado o conjunto de respostas adequadas a q em I com respeito a D.

Quanto ao mapeamento, os sistemas de integração $I = \langle G, S, M \rangle$ dividem-se em duas abordagens quanto ao foco do mapeamento. Num sistema de integração baseado na abordagem LAV (Local como Visão), os mapeamentos em M associam cada elemento s presente em S a uma consulta q_G sobre G (ou, a linguagem $L_{M,S}$ é restrita a descrever apenas elementos s). Já naqueles baseados na abordagem GAV (Global como Visão), as associações são entre elementos g do esquema global G e consultas q_S sobre a origem S.

Cada abordagem tem vantagens e características que a tornam própria para determinados problemas de integração; em particular, a construção de consultas em função de elementos de S ou G transfere a complexidade do mapeamento para este ou aquele esquema. No caso da abordagem LAV, a inclusão, remoção e edição de fontes de dados a S é mais fácil, pois cada fonte está descrita isoladamente das outras. Porém, a modificação do esquema global G implica em modificações em todos os mapeamentos. A abordagem GAV traz vantagens para sistemas onde as fontes de dados são estáveis mas o esquema global será modificado com mais frequência.

Ao elaborar consultas q_G para I, um sistema GAV, por ter o mapeamento feito entre elementos de g para consultas q_S em S, torna-se um problema de expansão de visões comum a bancos de dados relacionais, que é

resolvido e otimizado facilmente. Porém, num sistema LAV, onde os mapeamentos são feitos "ao contrário" entre $q_S \rightsquigarrow q_G$, existem duas técnicas possíveis, e ambas são difíceis do ponto de vista de complexidade computacional.

Uma delas é a reescrita de consultas baseada em visões (*view-based query rewriting*), onde dado uma consulta q_G e um conjunto de definições de visões, a consulta é reescrita de forma que referencie as visões e portanto acesse os dados em S. Certas formulações de visões e consultas podem tornar o problema de reescrita impossível de se chegar nas consultas originais; nestes casos, o objetivo passa a ser obter consultas aproximadas que retornem o resultado mais próximo das consultas originais possível. A segunda técnica para encontrar resultados a uma query q_G num sistema LAV é a de resposta a consultas baseada em visões (*view-based query answering*). Nesta técnica, conhecemos as extensões às visões definidas, e podemos inferir os resultados à consulta q_G verificando se, para cada tupla t existente nas extensões, segue-se logicamente que t está ou não em q_G.

Ambas técnicas resultam em grande complexidade computacional ao se utilizar graus moderados de expressividade, tanto nas consultas q_G quanto nas visões de mapeamento M. Reescrita de consultas baseada em visões, para consultas e visões expressas apenas com igualdades e conjunção de conceitos (sem utilização de negação e união, por exemplo) pertence à classe P de complexidade, ou seja, é relativamente eficiente; ao adicionar desigualdades, já pula para a classe co-NP, ou o tempo gasto para resolver o problema cresce exponencialmente com o tamanho do mesmo.

Anexo 2 - Otimização e Planejamento

Otimização é a etapa que identifica construções redundantes e as elimina, evitando repetição de trabalho. Uma consulta composta por diversas cláusulas terá o mesmo resultado, não importando a ordem em que são resolvidas. Porém, diferentes ordens de execução podem influenciar decisivamente na velocidade de resposta e quantidade de memória necessária. Portanto, na etapa de *Planejamento*, a consulta tem suas cláusulas reordenadas para que a execução seja feita no menor tempo possível. Implementamos mecanismos simples de otimização e planejamento de consultas, levando em conta o problema específico que tentamos resolver: uma consulta a diferentes *endpoints* possivelmente extensa (devido à ação do algoritmo de expansão).

OTIMIZAÇÃO

Acessar dados externos por meio da palavra-chave `SERVICE` implica em abrir uma conexão com o *endpoint*, enviar a consulta, aguardar a resolução da consulta SQL e receber os resultados. São dois os pontos de atenção quando uma consulta externa é feita:

- *Tempo*: Além do tempo gasto em comunicação com o *endpoint* remoto, a consulta SQL que será executada poderá demorar mais ou menos tempo, dependendo da complexidade da mesma e do uso concorrente do banco de dados;
- *Memória*: se a consulta externa retorna uma grande quantidade de resultados, isso implica tanto em maior demora para transferência mas também em maior uso de memória do computador, para abrigar os resultados.

Assim, a etapa de otimização implementada consiste em minimizar o número de chamadas `SERVICE`, agregando o máximo possível de cláusulas

buscadas no mesmo *endpoint*. As transformações implementadas baseam-se na associatividade das operações de união e intersecção, estão descritas abaixo e ilustradas na Figura 14.

```
SERVICE (E1) { A }
UNION SERVICE (E1) { B }
```
$\rightarrow$
```
SERVICE(E1){ A UNION B }
```

```
SERVICE(E1){ A }
UNION { SERVICE(E1){ B }
  UNION SERVICE(E2) { C }
}
```
$\rightarrow$
```
SERVICE(E1){ A UNION B }
UNION SERVICE(E2) { C }
```

Foram implementadas regras semelhantes para `JOIN`, bastando-se substituir a palavra-chave `UNION` nas regras acima.

PLANEJAMENTO

A etapa de planejamento consiste em executar os componentes de uma consulta em uma ordem que otimize a necessidade de memória e minimize o tempo de execução. Para cada nó da árvore sintática da consulta a ser planejada, é atribuído um valor de *custo*, que é estimado a partir dos outros nós vinculados a esse. Abaixo, as simplificações que utilizamos para estimativa de custo:

- O custo de uma operação `A JOIN B` é igual ao custo da operação `A`. Isto pois a operação `B` é executada em função dos resultados de `A`, o que pode ser muito melhor ou pior que executar a operação `B` isoladamente.
- O custo de uma operação `A UNION B` é igual à soma do custos de `A` e `B`. Isto porque ambas operações deverão ser executadas para realização da união.

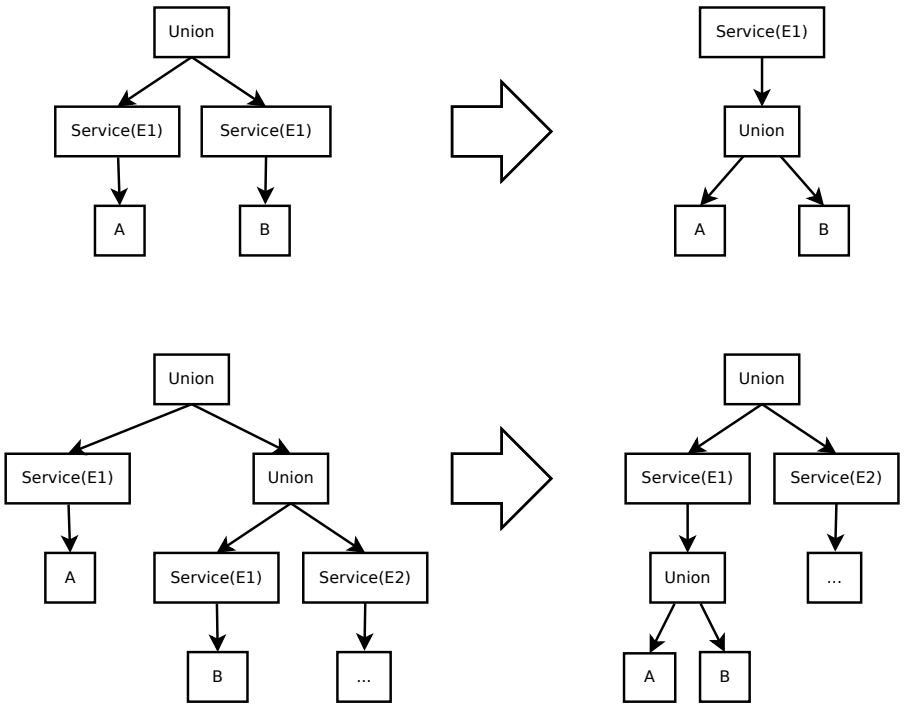

Figura 14 – Otimização de consultas federadas.

- O custo de uma operação `SERVICE(Ei){ ?x a C }`, onde `Ei` é um *endpoint* e `C` é uma classe, é determinado de maneira experimental, descrita a seguir.

Considerando-se que `Ei` é um *endpoint* OBDA, ou seja, que utiliza um banco de dados relacional para obtenção dos dados, não é trivial estimar o tempo de execução mesmo de uma consulta tão simples quanto

```
SELECT * { SERVICE(Ei) { ?x a C } }
```

Portanto, para planejar a execução da consulta federada de maneira eficaz, nosso algoritmo leva em consideração uma tabela de custos, relacionando cada *endpoint* e classe consultada a um tempo de execução e número de resultados. Sendo $\mathcal{C}$ o conjunto de todas as classes definidas na ontologia, $\mathcal{E}$ o conjunto de todos os *endpoints* utilizados, $n(E, C)$ o número de resultados retornados ao se consultar a classe C no *endpoint* E e $t(E, C)$ o tempo empregado, o custo será dado por:

$$cost(E, C) = 100 \left(\frac{n(E, C)}{2S_n} + \frac{t(E, C)}{2S_t} \right)$$

Os fatores S_n e S_t são iguais à soma do número de resultados e do tempo empregado para todas as classes e todos os *endpoints* da Federação:

$$S_n = \sum_{C \in \mathcal{C}} \sum_{E \in \mathcal{E}} n(E, C)$$

$$S_t = \sum_{C \in \mathcal{C}} \sum_{E \in \mathcal{E}} t(E, C)$$

A fórmula de custo resulta em um valor entre 0 a 100, sendo que valores menores de custo indicam que a consulta é mais rápida e retorna menos resultados. Implementamos um algoritmo simples para planejar a consulta, que reordena componentes unidos por `JOIN` de maneira que o primeiro componente avaliado tenha menor custo que o segundo.

Anexo 3 - Terminologias Médicas

CID-10

O CID-10 origina-se de uma codificação criada em 1893 é atualmente mantido pela Organização Mundial de Saúde (OMS) e tem mais de 14.400 códigos para doenças, achados anormais, queixas, circunstâncias sociais e causas externas de óbito. O sítio da Organização Mundial de Saúde o define como “ferramenta padrão de diagnóstico para epidemiologia, gerenciamento de saúde e propósitos clínicos”[10]. Foi homologado pela OMS em 1990 na 43a. Assembléia Mundial de Saúde e desde então está em uso em mais de 100 países e é citado em mais de 20.000 artigos científicos[11]. Atualmente, está em desenvolvimento o CID-11, cuja data de finalização prevista é 2015. O CID-11, ao contrário das versões anteriores, está sendo desenvolvido por um esforço cooperativo de diversos profissionais de saúde, informática e estatística, com o objetivo de gerar um sistema de codificação mais acurado e que atenda às necessidades de extração de informação de todos estes profissionais[12].

No Brasil, é utilizado como código para autorização de procedimentos em convênios, juntamente com o código do mesmo os avaliadores decidirão sobre a pertinência e cobertura do procedimento para a doença especificada. É também utilizada para a geração de estatísticas oficiais de causas de óbito, incidência e prevalência de doenças.

Os conceitos são organizados por capítulos, que contém um título descrevendo os tipos de doenças ou condições incluídos, os tipos excluídos, e outras notas para auxiliar a codificação. Os capítulos contém doenças (I - XIV), condições relacionadas à gravidez e puerpério (XV-XVI), má formação

[10] http://www.who.int/classifications/icd/en/

[11] http://www.who.int/classifications/icd/factsheet/en/index.html

[12] http://www.who.int/classifications/icd/revision/icd11faq/en/index.html

congênitas, deformidades e anormalidades cromossômica (XVII), sinais, sintomas e achados clínicos e laboratoriais anormais (XVIII), causas externas de morbidade e mortalidade (XIX-XX), fatores que provocam procura de serviços de saúde (XXI) e códigos de propósito especial (XXII).

Cada capítulo é dividido em blocos, com um grau um pouco maior de detalhamento dos tipos de doenças e condições. No capítulo relacionado a neoplasias (II), há quatro blocos principais, de neoplasias malignas, neoplasias in situ, neoplasias benignas e neoplasias de comportamento incerto ou desconhecido. O bloco de neoplasias malignas tem mais dois níveis de subdivisão, o primeiro separando neoplasias malignas pela definição do sítio primário (bem definido, mal definido, sistêmicos e de múltiplas localizações), e o segundo nível detalhando a subdivisão de neoplasias malignas com sítio bem definido em grandes regiões anatômicas: cabeça e pescoço, órgãos digestivos, intratorácicos, ossos e cartilagens, pele, mesotélio e tecido mole, mama, órgãos genitais masculinos, femininos, trato urinário, sistema nervoso central, e glândulas endócrinas.

Os códigos são compostos por uma letra seguida de dois números, e correspondem a uma categoria de doença. Um quarto número (separado por ponto dos anteriores) define o último nível de detalhe (em particular, ".9" indica que a doença não tem outra especificação). Para alguns propósitos específicos, este códigos não oferecem detalhamento suficiente, o que é explicado pela sua finalidade de prover dados para estatísticas globais. Este sistema permite a extensão arbitrária de seu código, pela adição de dígitos próprios após o quarto. Assim, se se deseja detalhar os tipos histológicos de câncer de mama do quadrante superior esquerdo (cujo código é C50.3), basta definir que o quinto dígito significará o subtipo, no nosso exemplo, "1" pode significar câncer ductal invasivo, "2" câncer lobular invasivo e assim por diante. Desta maneira, o código customizado permanece compatível com o original, bastando-se considerar apenas até o quarto dígito.

No texto original, é comum a indicação de sinônimos para alguns códigos, condições de exceção e outras restrições. No exemplo abaixo, neoplasias malignas de órgãos respiratórios e intratorácicos (C30-C39) inclui ouvido médio e exclui os mesoteliomas, que devem ser categorizados sob o código C45.-. Para cada conceito, são exibidos os sinônimos e conceitos excluídos.

Existem numerosas especializações do CID-10. Nos EUA, é comum o uso do ICD-10-CM (Clinical Modification), que extende as classificações do CID-10 para mais dois dígitos, totalizando códigos de até 6 dígitos, totalizando 68.000 códigos, contra 14.000 do CID-10 original. Há uma especialização do CID-10 para oncologia, o CID-O, descrito posteriormente. A ILDS (International League of Dermatological Societies) desenvolveu uma extensão do CID-10 para doenças dermatológicas. É resultado de uma cooperação entre um grupo de trabalho da sociedade de dermatologia alemã e a associação britânica de dermatologistas desde 1998. O ICECI - Classificação Internacional de Causas Externas de Ferimentos é complementar ao capítulo XX do CID-10, porém utiliza um sistema de classificação diferente. Permite codificar os objetos ou substâncias envolvidos nas ocorrências independentemente de outros ítens (como intenção, por exemplo). [13]

SNOMED CT

O SNOMED CT - Systematized Nomenclature of Medicine - Clinical Terms Spackman (2004) é um vocabulário que tem como objetivo descrever todos os conceitos utilizados em um prontuário. Contempla partes anatômicas, procedimentos e aspectos de estilo de vida relacionados à saúde. O SNOMED-CT é o resultado da integração dos Read Codes (UK) e o SNOMED-RT (Reference Terminology) (Spackman et al. 1997), e este foi desenvolvido a partir de 1965 pelo Colégio de Patologistas Americanos até 2007, quando o

[13] (http://www.rivm.nl/who-fic/ICECI/ICECI_1-2_2004July.pdf)

vocabulário foi adotado pela IHTSDO - International Health Terminology Standards Development Organization - uma organização sem fins lucrativos sediada na Dinamarca. São 310.000 conceitos organizados hierarquicamente. Os grupos superiores são: achado clínico, produto farmacêutico/biológico, organismo, contexto social e conceito de ligação.

Os conceitos são identificados por códigos numéricos e seus nomes completos, acrescidos de termos sinônimos e em alguns casos definições em texto livre. Além disso, há 50 tipos de conceitos de ligação, que vinculam conceitos entre si. Seus termos podem ser utilizados em uma linguagem de representação ontológica compatível com OWL-DL, na chamada pré ou pós coordenação de conceitos. A codificação de morfologias em câncer utilizada no SNOMED-CT é idêntica à utilizada no CID-O.

DSM-IV

O Diagnostic and Statistical Manual of Mental Disorders (DSM) é publicado pela Associação Psiquiátrica Americana, e descreve um vocabulário padronizado com classificação de transtornos mentais. Enquanto outras especialidades descrevem seus protocolos de diagnóstico em diferentes publicações, o DSM inclui, junto com o vocabulário, os critérios para diagnóstico. Sua primeira versão data de 1952, baseado em um esquema de classificação utilizado pelo exército americano para avaliação de seus soldados, e descrevia 106 transtornos mentais Grob (1991).

O DSM-IV possui mais de 500 códigos para diagnósticos psiquiátricos e utiliza cinco eixos de classificação, cada qual se referindo a um diferente aspecto: transtornos gerais, transtornos de personalidade e retardo mental, condições médicas gerais (potencialmente relevantes para a doença psiquiátrica), fatores psicossociais e ambientais, e avaliação geral de funcionalidade.

O código CID-9, em uso em 1987 quando o DSM-IV foi lançado, aparece vinculando a maior parte dos transtornos classificados, nos começos de seções descrevendo um transtorno e no conjunto de critérios para diagnóstico (DSM-IV). Alguns diagnósticos possuem subtipos, que são mutuamente exclusivos, e especificadores, instruções para o psiquiatra incluir junto do diagnóstico alguma particularidade aplicável ao mesmo. A versão mais atual é a DSM-5, aprovada pelo conselho curador da Associação Psiquiátrica Americana em 1o de dezembro de 2012. Algumas das principais mudanças com relação à versão anterior são a exclusão de variantes de esquizofrenia e a remoção da síndrome de Asperger [14].

READ CODES (STANDARD TERMINOLOGY ON UK)

A primeira versão dos códigos Read foi lançada em 1983 e contava com 30.000 termos codificados com 4 bytes (apropriado tendo em vista a capacidade de memória dos computadores da época). A estrutura do código era similar à do CID-9: o primeiro caractere indica o primeiro nível de categorização do termo, o segundo é relativo ao primeiro e assim por diante. São permitidos letras maiúsculas (com exceção do I e O) e minúsculas (exceto no primeiro nível), e números. Versões posteriores acrescentaram um byte à descrição e ampliaram para mais de 200.000 termos descritos em 1994. Fundiu-se ao SNOMED, gerando o SNOMED-CT em 1999, e o sistema de saúde inglês está atualmente migrando seus sistemas de informação dos Read Codes para o SNOMED-CT (de Lusignan 2005).

[14] http://www.dsm5.org/Documents/changes%20from%20dsm-iv-tr%20to%20dsm-5.pdf

OMIM

O Online Mendelian Inheritance on Men (OMIM) é uma base de dados sobre doenças, transtornos ou características humanas com componentes genéticos. Foi publicado a primeira vez em 1966 (McKusick 1998) e publicado em forma eletrônica desde então, na 12a edição em 1998. Está disponível online desde 1987. Atualmente tem mais de 15.000 entradas, cada qual correspondente a uma mutação, variação ou fenótipo. As entradas são codificadas através de um número de seis dígitos numéricos, sendo que o primeiro indica o tipo de herança relacionado: autossômica, vinculada ao cromossomo X, vinculada ao cromossomo Y ou mitocondrial. Além disso, há um caractere antes do código que indica se ela está relacionada a um gene, a um fenótipo ou ambos (Hamosh et al. 2005).

ICD-10-PCS

O ICD-10-PCS (Procedure Coding System) é um sistema de classificação para procedimentos desenvolvido nos EUA. Tem este nome pois é baseado nos códigos de procedimento do CID-9 e que foram descontinuados na versão oficial do CID-10. Utiliza códigos de 7 caracteres 0-9 e letras A-Z, exceto O e I (para evitar confusão com o zero e um). O primeiro caractere denota o tipo do procedimento (Médico/Cirurgico, Obstétrico, etc); os próximos dependem do primeiro código. Por exemplo, para procedimentos médicos, cirurgicos, os digitos seguintes denotam: sistema do corpo, operação, parte do corpo, abordagem, dispositivo e qualificador. Conta com 72.081 códigos, sua primeira versão foi desenvolvida pela 3M Health Information Systems e publicada em 1998, e é atualizada anualmente desde então.

TABELA DE PROCEDIMENTOS SUS

Desde 2008, os procedimentos, medicamentos, suprimentos e materiais especiais utilizados no Sistema Único de Saúde (SUS) constam de uma terminologia unificada chamada Tabela de Procedimentos, Medicamentos e OPM do SUS. A tabela é gerenciada exclusivamente pela Secretaria de Atenção à Saúde, por meio do Departamento de Regulação, Avaliação e Controle de sistemas, de acordo com a portaria GM/MS no. 2.848 de 06 de novembro de 2007. Esta terminologia é organizada hierarquicamente, com quatro níveis de profundidade. O primeiro nível é chamado grupo, que identifica os procedimentos pela finalidade do atendimento ou que tenham características gerais similares. O segundo organiza os procedimentos por tipo ou área de atuação. O terceiro organiza por diferentes critérios, podendo ser a região anatômica ou sistema do corpo humano no qual age o procedimento, a especialidade médica envolvida, o tipo de exame, órtese, prótese ou cirurgia. Por fim, no quarto nível da hierarquia estão os procedimentos os materiais. O código identificador de cada ítem é composto por 5 blocos numéricos, sendo os quatro primeiros identificadores dos diferentes níveis da hierarquia e o último um dígito verificador para evitar erros de digitação. Para cada ítem da tabela, existem atributos como a complexidade do procedimento, o perfil do paciente (sexo, idade, CID principais e secundários, tempo de permanência) e informações administrativas (vigência, valor do procedimento, equipe médica envolvida, habilitação do estabelecimento para realização do procedimento) e vínculos com as tabelas anteriores (SIA e SIH). Estas tabelas são disponibilizadas livremente, assim como o sistema SIGTAP para consulta e visualização das mesmas.

CBHPM

A Classificação Brasileira Hierarquizada de Procedimentos Médicos (CBHPM) é mantida pela Associação Médica Brasileira (AMB), e é uma terminologia dos procedimentos médicos realizados no Brasil, organizada em quatro níveis. O primeiro nível da hierarquia é chamado Capítulo, e contempla procedimentos gerais, clínicos, cirúrgicos e invasivos e diagnósticos e terapêuticos. Dentro de cada capítulo existem seções, organizando os capítulos por subtipo de procedimento, região anatômica ou sistemas do corpo humano. Em cada seção há subseções, restringindo ainda mais os procedimentos por critérios variados, e em seguida os procedimentos. Cada procedimento tem um código composto por cinco blocos numéricos, cada um organizado por um dos níveis da hierarquia, e o último um dígito de verificação. A finalidade desta classificação não é atribuir valores mínimos ou referenciais para cada procedimento, mas prover uma ordenação dos procedimentos por complexidade técnica. Além do porte, cada procedimento tem uma descrição, e alguns especificam custo operacional, número de auxiliares e porte anestésico.

TUSS

A Terminologia Unificada de Saúde Suplementar (TUSS) foi instituída pela Instrução Normativa no 34/2009 da Diretoria de Desenvolvimento Setorial da Agência Nacional de Saúde (ANS) e criada pelo Comitê de Padronização das Informações em Saúde Suplementar (COPISS), com membros da equipe técnica da ANS e da Associação Médica Brasileira (AMB). Tem como base a terminologia CBHPM, porém seu foco é nos procedimentos de quaisquer especialidade da área de saúde (enquanto a outra se restringe aos efetuados por médicos) e no faturamento destes procedimentos. O código identificador

de procedimentos é idêntico ao da CBHPM.

ROL DE PROCEDIMENTOS

Tem como objetivo listar os serviços mínimos que os planos de saúde (ou plano-referência) devem atender no Brasil. O Rol não dispõe de códigos para seus termos, ao invés disso lista os tipos de serviço que devem ser contemplados para posterior vínculo aos códigos da CHBPM. Sua primeira versão foi publicada em 2001 e sua gestão é de responsabilidade da ANS.

CPT/HCPCS

Há nos Estados Unidos da América um programa público de seguridade social chamado Medicare, que provê acesso a serviços de saúde para maiores de 65 anos de idade, pessoas com necessidades especiais e casos severos de falência renal crônica. Para padronizar a terminologia utilizada nas trocas de informação entre serviços de saúde e o Medicare, foi criado o Healthcare Common Procedure Coding System (HCPCS). Ele é dividido em dois principais subsistemas, conhecidos como Nível I e Nível II. O Nível I da terminologia compreende termos descritivos e códigos numéricos que identificam serviços e procedimentos médicos que serão faturados contra convênios e o Medicare. Esta terminologia é mantida pela Associação Médica Americana (AMA). O Nível II compreende produtos, suprimentos e serviços não incluídos no Nível I, tais como serviços de remoção, próteses, órteses, equipamentos médicos duráveis e suprimentos utilizados fora de um consultório médico. Os códigos são sequenciais e organizados por diagnóstico, porém não há organização hierárquica.

ATC (WHO)

Com o objetivo de realizar pesquisa sobre utilização de drogas para melhorar a qualidade de seu uso, a Organização Mundial de Saúde mantém o Anatomic Therapeutic Chemical - ATC - uma terminologia sobre drogas medicinais. Toda droga é organizada de acordo com 5 níveis hierárquicos: no primeiro, o órgão ou sistema em que a droga age (Anatomic), no segundo, a terapêutica a que se propõe (Therapeutic), no terceiro e quarto nível os subgrupos químicos/farmacológicos/terapêuticos e no quinto nível a substância química. Cada droga pode ser classificada em diferentes ramos, caso seja empregada em mais de uma situação. Por exemplo, Acido Acetilsalicílico pode ser empregado como inibidor de agregação de plaquetas (sendo classificado sob o grupo anatômico B- Blood and blood forming organs) ou como analgésico (sob o grupo N - Nervous system).

NDC (National Drug Code - FDA) O National Drug Code (NDC) é um registro de drogas utilizadas em seres humanos mantido pelo US Food and Drug Administration (FDA). Utiliza 10 dígitos numéricos divididos em três blocos (de 1 ou mais dígitos cada, mas sempre totalizando 10 algarismos) que identificam o fabricante, o produto e o formato de empacotamento. Para cada entrada neste registro, há dados do nome do produto, nome da substância ativa, o tipo de produto, forma de aplicação, e data de início de divulgação.

CID-O

O CID-O 3a edição é considerado uma adaptação do CID-10 para oncologia. Nesta codificação, criada em 1976 e atualizada pela última vez em 2000, os tumores são classificados em um sistema de quatro eixos, contemplando sítio, morfologia, comportamento e graduação. A classificação de sítio

é feita utilizando-se os códigos C00 até C99, com algumas mudanças (por exemplo, câncer de pele utiliza apenas o código C44, e o código morfológico distingue entre melanoma e não-melanoma). O código morfológico é iniciado com a letra M, quatro dígitos que determinam o tipo histológico, então uma barra "/", um quarto dígito indicando o comportamento (benigno, in situ, invasor) e o quinto para o grau de diferenciação. Os códigos morfológicos são organizados em categorias, cada qual pode corresponder a um tipo específico de sítio.

TNM

O sistema TNM de estadiamento de câncer nasceu da observação de que os casos localizados da doença, isto é, casos em que o câncer não havia se espalhado para outros órgãos, apresentavam taxas de sobrevida maiores do que aqueles em que a doença já havia se estendido. Este sistema tem como objetivo ajudar o médico a planejar o tratamento, ter uma indicação do prognóstico, auxiliar a interpretação dos resultados do tratamento, facilitar a troca de informações entre centros de tratamento e contribuir para a pesquisa sobre o câncer (TNM 6a edição). As variáveis avaliadas pelo médico são três, a T, que reflete a extensão do tumor primário, a N, extensão de metástase em linfonodos regionais e a M, extensão de metástase em órgãos a distância. A variável T recebe valores inteiros de 0 a 4, N de 0 a 3, e M recebe 0 ou 1. Dependendo da localização, letras de a-d podem ser adicionadas ao final do código, para criação de subcategorias. Além disso, o tipo de informação utilizada para a classificação (em consulta clínica ou com exame anatomopatológico), o momento do diagnóstico (antes, durante/após o tratamento, após uma recidiva ou por autópsia), a presença de múltiplos tumores numa mesma localização também podem ser representados com letras adicionais. Outros símbolos opcionais descrevem invasão linfática, invasão venosa,

graduação histopatológica, localização das metástases, presença de células tumorais isoladas, status de invasão do linfonodo sentinela, o grau de certeza da informação obtida e a presença de tumor residual após o tratamento.

Para cada localização, existe uma série de regras para estadiamento da doença que permite condensar as variáveis TNM e apenas uma, o estadio, que pode ser 0 (no caso de tumores in situ) até IV (no caso de tumores com metástase à distância). Assim como nas variáveis TNM, em algumas localizações o estadio pode receber uma letra adicional. As regras de agrupamento seguidas diferem entre localizações, porém as sobrevidas de cânceres diferentes para um mesmo estadio devem ser compatíveis.

NCI THESAURUS

O National Cancer Institute Thesaurus (NCIt) é uma terminologia de referência, cobrindo o domínio de atendimento clínico, pesquisa básica e translacional, dados públicos e atividades administrativas. Seu conteúdo é proveniente do próprio National Cancer Institute (NCI) e de parceiros como o US Food and Drug Administration (FDA), e é baseado em lógica de descrição (Kumar e Smith 2005), implementando relações entre conceitos semanticamente ricas. O NCI Thesaurus é baseado no NCI Metathesaurus, que por sua vez originou-se no UMLS (será visto a seguir). Seu objetivo inicial era apoiar o registro e obtenção de informações acuradas sobre atividades relacionadas à missão de desenvolvimento científico do instituto. É codificada em OWL Lite, que é um subconjunto de OWL DL que contempla a complexidade necessária para representar a ontologia.

MESH

O Medical Subject Headings (MeSH) é um vocabulário controlado composto por termos utilizados na indexação de conteúdo de documentos de medicina e áreas correlatas. Em particular, é utilizado para indexar os resumos presentes no MEDLINE (REF), com mais de 10 milhões de citações. Seus termos obedecem a hierarquias múltiplas, onde um termo pode pertencer a uma ou mais super categorias. Possui mais de 25.000 termos e 213.000 termos de entrada (que podem incluir sinônimos), e está disponível em 41 idiomas.

MEDDRA

O Medical Dictionary for Regulatory Activities (MedDRA) foi criado pela Conferência Internacional em Harmonização (ICH[15]) para padronizar o registro e notificação de eventos adversos em estudos clínicos. Os conceitos médicos estão organizados em uma hierarquia de cinco níveis, onde o primeiro é chamado System Organ Class (SOC), que indica um sistema físico ou fisiológico, etiologia, ou propósito. O segundo nível contém conceitos do tipo High Level Group Terms (HLGT), que agrupa um ou mais High Level Terms (HLT). No quarto nível, temos os Preferred Terms (PT), que por sua vez contém os respectivos sinônimos ou Lower Level Terms. Estes termos definem desde sinais (como ansiedade - id: 10001497), doenças (como falência cardíaca - id: 10007554) até procedimentos (como amniocentese - 10001958).

Para facilitar a busca por relatórios de caso potencialmente relevantes, foram criados os Standardised MedDRA Queries (SMQ), que são agrupamentos de conceitos (preferencialmente no nível dos PTs). Cada SMQ é composto por um título (por exemplo, Acute Central Respiratory Depression), com a definição dividida em tópicos descritivos, critérios de inclusão e exclusão onde

[15] http://www.ich.org/

efetivamente são utilizados os PTs relevantes para atender ao critério, notas e referências.

LOINC

Com o intuito de padronizar a terminologia utilizada em documentos eletrônicos para representar informação clínica (como resultados de testes de laboratório), o Regenstrief Institute, uma organização médica sem fins lucrativos sediada em Indiana, EUA, criou em 1994 o Logical Observation Identifier Names and Codes (LOINC) e é responsável por sua manutenção desde então. Os conceitos são identificados com uma sequencia de 5 dígitos sequenciais, um traço e um dígito de verificação redundante. Os nomes dos conceitos são organizados em diversos componentes separados por dois pontos: componente analisado, sua propriedade (concentração, massa, volume), a duração do experimento, o tipo de amostra (soro, urina, etc), a escala de medida (quantitativa ou qualitativa) e o método de análise. Apesar de os conceitos serem organizados hierarquicamente, e isto estar disponível em alguns softwares, não são disponibilizados dados oficiais para reconstruir essa organização. Porém, é possível consultar a hierarquia de conceitos no site do BioPortal [16].

UMLS

As terminologias discutidas nesta tese foram escolhidas por sua relevância no contexto de informática médica no mundo, no Brasil e para o trabalho desenvolvido, representando uma ínfima parcela das terminologias e ontologias existentes em medicina e enfermagem. Esta profusão já em 1986 levou a National Library of Medicine (NLM) dos EUA a criar uma biblioteca

[16] (http://bioportal.bioontology.org/ontologies/1350)

unificada com as principais terminologias, classificações e ontologias biomédicas, e mapeamento entre os termos das mesmas. Atualmente, são mais de 100 vocabulários, 1 milhão de conceitos, 2 milhões de sinônimos e 12 milhões de relacionamentos entre conceitos, que podem conter anotações em diversas linguagens. Por conter diferentes terminologias que tem diferentes licenças de copyright, o uso de cada terminologia está sujeito a regras próprias, apesar de estarem incluídas no pacote.

HL7 MESSAGING

Com a crescente informatização que a área de saúde tem sofrido desde a década de 80, a troca de informação entre serviços de saúde, órgãos governamentais e operadoras de saúde depende de envio de arquivos de computador entre estas instituições. Com cada uma destas instiuições utilizando um sistema de informação independente, logo os fabricantes de software perceberam a necessidade de se estabelecer um padrão para estas comunicações. Um destes padrões é o Health Level 7 v2 (HL7v2) que define a linguagem, estrutura e tipos de dados para mensagens entre sistemas, facilitando a interoperação destes. O HL7v2 foi estabelecido em 1989, e a versão 3 (HL7v3) começou a ser desenvolvida em 1995 e foi publicada em 2005, já utilizando arquivos em formato XML.

HL7 RIM

A grande complexidade dos processos hospitalares e da informação médica torna complicada a construção de sistemas de informação adequados. Grande parte deste problema é a modelagem dos dados, que se feita de forma inapropriada irá atrapalhar a expansão e manutenção do sistema de informação, limitando sua utilidade. O Health Level 7 Reference Information

Model (HL7 RIM) é um modelo padrão de dados para implementação de fluxos clínicos e administrativos em um sistema de saúde. Esta modelagem é feita em um modelo orientado a objetos, e representado em Unified Modeling Language (UML), disponibilizado sem custos.

TISS

Estabelecida pela ANS para padronizar a troca de informações entre operadoras de saúde e prestadores de serviços em saúde, a Troca de Informações em Saúde Suplementar (TISS). Utiliza-se do formato XML de arquivos que está descrito em documentação provida pela ANS. As trocas de informação contempladas pelo TISS compreendem autorização de procedimentos, comunicação de internação ou alta, recurso de glosa, cobrança de serviços de saúde, dentre outros. Especifica por fim as terminologias utilizadas em seus componentes, como o TUSS e o CID-10.

OPENEHR

Enquanto o HL7 RIM define uma modelagem de dados para representar processos hospitalares, o armazenamento de informações clínicas não é contemplado. A fundação sem fins lucrativos OpenEHR Foundation mantem o OpenEHR, uma especificação de padrão livre sobre gerenciamento, armazenamento, recuperação e troca de informação em saúde em prontuários eletrônicos do paciente.

O modelo de dados contempla aspectos de gerenciamento de documentos quanto ao PEP, porém trata como uma entidade à parte a definição de informação clínica, separando assim aspectos que tem diferentes graus de estabilidade. O gerenciamento de documentos se ocupa em relacionar um

documento a um encontro com serviço médico (ou seja, uma consulta, internação, cirurgia, etc), uma data, cuidar de sua assinatura eletrônica e outras preocupações de cunho gerencial. A informação clínica, que está efetivamente contida nestes documentos, constantemente muda e evolui, refletindo as mudanças no conhecimento médico, no processo e outras peculiaridades de uma instituição, ou mesmo na especialidade médica do usuário.

Para flexibilizar ao máximo a informação clínica sem necessitar alterações no modelo de referência, existem dois artefatos a parte deste último para representar esta informação: arquétipos e templates. Arquétipos são modelos de perguntas e respostas, desenvolvidos e mantidos de maneira aberta, organizados hierarquicamente por tema. Templates são modelos de documentos eletrônicos que utilizam arquétipos para representar suas perguntas.

Anexo 4 - Mapeamentos

Neste anexo listamos todos os mapeamentos criados para a realização do experimento descrito no capítulo 5.

MAPEAMENTOS DO SISTEMA EHR

```
select "CD_PACIENTE" from paciente
```

Listagem A.1 – Consulta para mapeamento da classe Paciente no sistema EHR.

```
select "CD_PACIENTE" from paciente
 where "DT_NASCIMENTO" < now() - interval '18 year'
```

Listagem A.2 – Consulta para mapeamento da classe PacienteMaiorDe18Anos no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime, pre_med, itpre_med
 where
 atendime."CD_ATENDIMENTO" = pre_med."CD_ATENDIMENTO"
 AND pre_med."CD_PRE_MED" = itpre_med."CD_PRE_MED"
 AND itpre_med."CD_TIP_PRESC" in ( 38210,38213 )
 ) tmp
```

Listagem A.3 – Consulta para mapeamento da classe Paclitaxel no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime, pre_med, itpre_med
 where
 atendime."CD_ATENDIMENTO" = pre_med."CD_ATENDIMENTO"
 AND pre_med."CD_PRE_MED" = itpre_med."CD_PRE_MED"
```

```
 AND itpre_med."CD_TIP_PRESC" = 38221
 ) tmp
```

Listagem A.4 – Consulta para mapeamento da classe Docetaxel no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime, pre_med, itpre_med
 where
 atendime."CD_ATENDIMENTO" = pre_med."CD_ATENDIMENTO"
 AND pre_med."CD_PRE_MED" = itpre_med."CD_PRE_MED"
 AND itpre_med."CD_TIP_PRESC" =38365
 ) tmp
```

Listagem A.5 – Consulta para mapeamento da classe Cetuximab no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime, pre_med, itpre_med
 where
 atendime."CD_ATENDIMENTO" = pre_med."CD_ATENDIMENTO"
 AND pre_med."CD_PRE_MED" = itpre_med."CD_PRE_MED"
 AND itpre_med."CD_TIP_PRESC"  in ( 47668, 48541, 48546 )
 ) tmp
```

Listagem A.6 – Consulta para mapeamento da classe Erlotinib no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime, pre_med, itpre_med
 where
 atendime."CD_ATENDIMENTO" = pre_med."CD_ATENDIMENTO"
```

```
 AND pre_med."CD_PRE_MED" = itpre_med."CD_PRE_MED"
 AND itpre_med."CD_TIP_PRESC"  =38368
 ) tmp
```

Listagem A.7 – Consulta para mapeamento da classe Trastuzumab no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime, pre_med, itpre_med
 where
 atendime."CD_ATENDIMENTO" = pre_med."CD_ATENDIMENTO"
 AND pre_med."CD_PRE_MED" = itpre_med."CD_PRE_MED"
 AND itpre_med."CD_TIP_PRESC"  = 48638
 ) tmp
```

Listagem A.8 – Consulta para mapeamento da classe Gefitinib no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime, pre_med, itpre_med
 where
 atendime."CD_ATENDIMENTO" = pre_med."CD_ATENDIMENTO"
 AND pre_med."CD_PRE_MED" = itpre_med."CD_PRE_MED"
 AND itpre_med."CD_TIP_PRESC" in ( 38210,38213 )
 group by "CD_PACIENTE"
 having max(DT_PRE_MED)  < now() - interval '1 month'
 ) tmp
```

Listagem A.9 – Consulta para mapeamento da classe TaxanoHaMaisDeQuatroSemanas no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
```

```
 from atendime
 where "CD_CID" like 'C50%' ) teste_c50
```

Listagem A.10 – Consulta para mapeamento da classe C50 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0150', 'ID0371', 'ID1411'
, 'ID1690', 'ID2035')
 and registro_resposta."DS_RESPOSTA" like '%0%'
 ) tmp
```

Listagem A.11 – Consulta para mapeamento da classe T0 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0150', 'ID0371', 'ID1411'
, 'ID1690', 'ID2035')
 and registro_resposta."DS_RESPOSTA" like '%1%'
 ) tmp
```

Listagem A.12 – Consulta para mapeamento da classe T1 no sistema EHR.

```
select "CD_PACIENTE" FROM (
 select distinct "CD_PACIENTE" from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0150', 'ID0371', 'ID1411'
, 'ID1690', 'ID2035')
 and registro_resposta."DS_RESPOSTA" like '%2%'
 ) tmp
```

Listagem A.13 – Consulta para mapeamento da classe T2 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0150', 'ID0371', 'ID1411'
, 'ID1690', 'ID2035')
 and registro_resposta."DS_RESPOSTA" like '%3%'
 ) tmp
```

Listagem A.14 – Consulta para mapeamento da classe T3 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0150', 'ID0371', 'ID1411'
, 'ID1690', 'ID2035')
 and registro_resposta."DS_RESPOSTA" like '%4%'
 ) tmp
```

Listagem A.15 – Consulta para mapeamento da classe T4 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0151', 'ID0372', 'ID1412'
, 'ID1691', 'ID2037')
 and registro_resposta."DS_RESPOSTA" like '%0%'
 ) tmp
```

Listagem A.16 – Consulta para mapeamento da classe N0 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
```

```
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0151', 'ID0372', 'ID1412'
, 'ID1691', 'ID2037')
 and registro_resposta."DS_RESPOSTA" like '%1%'
 ) tmp
```

Listagem A.17 – Consulta para mapeamento da classe N1 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0151', 'ID0372', 'ID1412'
, 'ID1691', 'ID2037')
 and registro_resposta."DS_RESPOSTA" like '%2%'
 ) tmp
```

Listagem A.18 – Consulta para mapeamento da classe N2 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
```

```
 pergunta_doc."DS_PERGUNTA" in ('ID0151', 'ID0372', 'ID1412'
, 'ID1691', 'ID2037')
 and registro_resposta."DS_RESPOSTA" like '%3%'
 ) tmp
```

Listagem A.19 – Consulta para mapeamento da classe N3 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0152', 'ID0373', 'ID1412'
, 'ID1692', 'ID2038')
 and registro_resposta."DS_RESPOSTA" like '%0%'
 ) tmp
```

Listagem A.20 – Consulta para mapeamento da classe M0 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" in ('ID0152', 'ID0373', 'ID1412'
, 'ID1692', 'ID2038')
 and registro_resposta."DS_RESPOSTA" like '%1%'
```

```
 ) tmp
```

Listagem A.21 – Consulta para mapeamento da classe M1 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0779'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.22 – Consulta para mapeamento da classe T0 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0780'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.23 – Consulta para mapeamento da classe T1 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
```

```
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0781'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.24 – Consulta para mapeamento da classe T2 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0782'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.25 – Consulta para mapeamento da classe T3 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
```

```
 pergunta_doc."DS_PERGUNTA" = 'ID0783'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.26 – Consulta para mapeamento da classe T4 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0786'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.27 – Consulta para mapeamento da classe N0 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0787'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.28 – Consulta para mapeamento da classe N1 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0788'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.29 – Consulta para mapeamento da classe N2 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0789'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.30 – Consulta para mapeamento da classe N3 no sistema EHR.

```
select "CD_PACIENTE" FROM ( select distinct "CD_PACIENTE"
 from atendime
 join   registro_documento using ( "CD_ATENDIMENTO" )
 join   registro_resposta using ( "CD_REGISTRO_DOCUMENTO" )
 join   pergunta_doc using ( "CD_PERGUNTA_DOC" )
```

```
 where
 pergunta_doc."DS_PERGUNTA" = 'ID0791'
 and registro_resposta."DS_RESPOSTA" = 'checked'
 ) tmp
```

Listagem A.31 – Consulta para mapeamento da classe M0 no sistema EHR.

MAPEAMENTOS DO SISTEMA AP

```
select rgh from ( select distinct rgh
 from laudos
 where (  laudos.texto ilike '%ADENOCARCINOMA%'
 or laudos.texto ilike '%CARCINOMA%adenoide%'
 or laudos.texto ilike '%CARCINOMA%tubular%'
 or laudos.texto ilike '%CARCINOMA%cribriforme%'
 or laudos.texto ilike '%CARCINOMA%ductal%' )
 ) tmp
```

Listagem A.32 – Consulta para mapeamento da classe AdenocarcinomaInvasivo no sistema AP.

```
select rgh from ( select distinct rgh
 from laudos
 where
 texto ilike '%IMUNOISTOQ%' AND
 lower(texto) ~ '(erb|her)[^\\:]*\\:[^\\:]*positivo.[^\\:123
]*(escore)?3(\\\\+)?'
 ) tmp
```

Listagem A.33 – Consulta para mapeamento da classe IHQ_HER2_Escore3 no sistema AP.

```
select rgh from ( select distinct rgh
 from laudos
 where
 texto ilike '%IMUNOISTOQ%' AND
 lower(texto) ~ '(erb|her)[^\\:]*\\:[^\\:]*positivo.[^\\:123
]*(escore)?2(\\\\+)?'
 ) tmp
```

Listagem A.34 – Consulta para mapeamento da classe IHQ_HER2_Escore2 no sistema AP.

```
select rgh from ( select distinct rgh
 from laudos
 where
 texto ilike '%IMUNOISTOQ%' AND
 lower(texto) ~ '(erb|her)[^\\:]*\\:[^\\:]*negativo.[^\\:023
]*(escore)?1(\\\\+)?'
 ) tmp
```

Listagem A.35 – Consulta para mapeamento da classe IHQ_HER2_Escore1 no sistema AP.

```
select rgh from ( select distinct rgh
 from laudos
 where
 texto ilike '%IMUNOISTOQ%' AND
 lower(texto) ~ '(erb|her)[^\\:]*\\:[^\\:]*negativo.[^\\:123
]*(escore)?0(\\\\+)?'
```

```
) tmp
```

Listagem A.36 – Consulta para mapeamento da classe IHQ_HER2_Escore0 no sistema AP.

MAPEAMENTOS DO SISTEMA RHC

```
select rgh from ( select distinct rgh from RHC
 where dtnasc + interval '18 year' < now() ) dist
```

Listagem A.37 – Consulta para mapeamento da classe PacienteMaiorDe18Anos no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where quimio like '1%' ) dist
```

Listagem A.38 – Consulta para mapeamento da classe Quimioterapia no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.morfo ~ '[3-6]$'  AND RHC.morfo between '814' and
 '83899' ) dist
```

Listagem A.39 – Consulta para mapeamento da classe AdenocarcinomaInvasivo no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 WHERE RHC.topo like 'C50%' ) dist
```

Listagem A.40 – Consulta para mapeamento da classe C50 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.T like '0%' ) dist
```

Listagem A.41 – Consulta para mapeamento da classe T0 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.T like '1%' ) dist
```

Listagem A.42 – Consulta para mapeamento da classe T1 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.T like '2%' ) dist
```

Listagem A.43 – Consulta para mapeamento da classe T2 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.T like '3%' ) dist
```

Listagem A.44 – Consulta para mapeamento da classe T3 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.T like '4%' ) dist
```

Listagem A.45 – Consulta para mapeamento da classe T4 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.N like '0%' ) dist
```

Listagem A.46 – Consulta para mapeamento da classe N0 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.N like '1%' ) dist
```

Listagem A.47 – Consulta para mapeamento da classe N1 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.N like '2%' ) dist
```

Listagem A.48 – Consulta para mapeamento da classe N2 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.N like '3%' ) dist
```

Listagem A.49 – Consulta para mapeamento da classe N3 no sistema RHC.

```
select rgh from ( select distinct rgh
 from RHC
 where RHC.M like '0%' ) dist
```

Listagem A.50 – Consulta para mapeamento da classe M0 no sistema RHC.

Neste Anexo descrevemos todas as regras que utilizam intersecção de classes feita no capítulo 5.

$$AdenocarcinomaInvasivo \sqcap DoencaMetastatica \sqcap HER2positivo \sqcap PacienteMaiorDe18Anos \sqcap Taxanos \sqsubseteq CriterioTriagem$$

$$AdenocarcinomaInvasivo \sqcap DoencaLocalAvancada \sqcap HER2positivo \sqcap PacienteMaiorDe18Anos \sqcap Taxanos \sqsubseteq CriterioTriagem$$

$$AdenocarcinomaInvasivo \sqcap DoencaMetastatica \sqcap HER2positivo \sqcap PacienteMaiorDe18Anos \sqcap Quimioterapia \sqsubseteq CriterioTriagemRelaxado$$

$$AdenocarcinomaInvasivo \sqcap DoencaLocalAvancada \sqcap HER2positivo \sqcap PacienteMaiorDe18Anos \sqcap Quimioterapia \sqsubseteq CriterioTriagemRelaxado$$

$$C50 \sqcap M0 \sqcap N0 \sqcap T4 \sqsubseteq C50\_ECIIIb$$

$$C50 \sqcap M0 \sqcap N1 \sqcap T4 \sqsubseteq C50\_ECIIIb$$

$$C50 \sqcap M0 \sqcap N2 \sqcap T4 \sqsubseteq C50\_ECIIIb$$

$$C50 \sqcap M0 \sqcap N3 \sqcap T4 \sqsubseteq C50\_ECIIIc$$

$$C50 \sqcap M1 \sqsubseteq C50\_ECIV$$

$$C50_E CIIIc \sqcap T4 \sqsubseteq DoencaLocalAvancada$$

$$IHQ\_HER2\_Escore2 \sqcap ISHRazaoHerCHR17Maior2.2 \sqsubseteq HER2positivo$$