Um estudo de desempenho de consultas em dados abertos em saúde utilizando ferramentas OBDA

Samuel L. B. Bispo
ra103643@uem.br
Universidade Estadual de Maringá
Maringá, Paraná, Brasil

Ana H. B. Mazur
ra118003@uem.br
Universidade Estadual de Maringá
Maringá, Paraná, Brasil

Vinicius K. Fukace
ra115672@uem.br
Universidade Estadual de Maringá
Maringá, Paraná, Brasil

Raqueline R. M. Penteado
rrmpenteado@uem.br
Universidade Estadual de Maringá
Maringá, Paraná, Brasil

Heloise M. P. Teixeira
hmpteixeira@uem.br
Universidade Estadual de Maringá
Maringá, Paraná, Brasil

ABSTRACT

With the growing amount of information managed by Information Systems presented by different institutions, data integration poses challenges, such as the execution of queries in multiple data sources with computational cost and viable response time. The present work describes a practical experiment, which had as main objective to verify the performance of a OBDA tool with different DBMS in the queries execution in SUS open data sources. The obtained results indicate that the choice of the database management system can influence the performance of the queries during the data integration process. Discussing the performance of technological tools and methods for integrating health data is a relevant topic to achieve the objectives of access to information distributed in the different Health Information Systems (HIS).

KEYWORDS

OBDA, Query Performance, DATASUS

1 INTRODUÇÃO

Sistemas de Informação em Saúde (SIS) armazenam e gerenciam dados provindos de diferentes serviços de saúde para gerar informações e conhecimento para tomada de decisão, tanto no âmbito administrativo como nos cuidados em saúde. Conforme [8], existem 54 diferentes SIS que são alimentados pelos municípios do país, sem considerar os do setor privado.

Em um cenário ideal, os SIS desenvolvidos com diferentes tecnologias e padrões deveriam se comunicar para integrar e processar informações valiosas, que podem gerar conhecimento para tomada de decisão clínica e administrativa em saúde. No entanto, um projeto de integração envolvendo tantos tipos de sistemas não é trivial, surgindo uma série de desafios para desenvolvedores, sendo um deles a execução de consultas processadas em múltiplas fontes de dados em um tempo de execução viável.

Uma abordagem para possibilitar a comunicação entre SIS é o desenvolvimento da padronização de dados, expressos por exemplo, em ontologias, que descrevem as informações de um domínio por meio de uma linguagem compreensível por sistemas computacionais. A utilização de ontologias para acesso a dados distribuídos é proposta no paradigma OBDA (Ontology Based Data Access) [20], que utiliza ontologias como camada conceitual sobre os dados, oferecendo uma descrição que facilita a execução de consultas em diferentes Sistemas Gerenciadores de Banco de Dados (SGBDs).

Essa camada melhora a expressividade dos dados mas gera um custo extra em termos de complexidade computacional, portanto, OBDA tem tradicionalmente sido um recurso caro, interessante quando os desafios a serem enfrentados compensam o custo da solução [11]. Neste contexto, um desafio no desenvolvimento de projetos de integração de SIS é selecionar e utilizar ferramentas e técnicas de integração de modo que tenham custo computacional viável.

Este trabalho apresenta um estudo sobre ferramentas OBDA para realizar consultas em dados abertos disponibilizados pelo Departamento de Informática do Sistema Único de Saúde do Brasil (DATASUS)[1], uma importante fonte de dados abertas que pode subsidiar análise e tomada de decisão baseada em evidência e elaboração de programas de ações em saúde. No entanto, a visualização e análise da grande quantidade de dados abertos disponibilizados pelo DATASUS para usuários sem conhecimento técnico não é uma tarefa simples, exigindo conhecimento das ferramentas usadas para obter, filtrar e analisar os dados. Desta grande quantidade de dados, quando processados por uma ferramenta computacional, pode-se extrair informações valiosas para a quantificação e avaliação das informações em saúde. Ferramentas como TabNet e TabWin (desenvolvidos pelo DATASUS) permitem gerar, organizar e consultar dados oficiais do Sistema Único de Saúde (SUS)[2]. Apesar desses aplicativos facilitarem o acesso aos dados, exigem conhecimento técnico para que o usuário consiga utilizar todos os seus recursos.

Neste contexto, apresenta-se um estudo de desempenho com a utilização de ferramentas para obtenção e integração de dados abertos em saúde. Discutir o desempenho no uso de ferramentas e métodos para integração de dados em saúde é um tema relevante para se atingir os objetivos de acesso à informação distribuída nos diferentes SIS. A principal contribuição da presente pesquisa é apresentar um experimento prático, que pode auxiliar desenvolvedores em projetos de integração de SIS que utilizem ferramentas OBDA.

As seções seguintes do artigo estão organizadas como segue: a seção 2 descreve os principais conceitos e tecnologias utilizadas, a seção 3 apresenta os procedimentos e ferramentas utilizadas, a seção 4 descreve os experimentos realizados e os resultados. Por fim, a seção 5 descreve as considerações finais.

[1] https://datasus.saude.gov.br/
[2] https://datasus.saude.gov.br/transferencia-de-arquivos/

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os principais conceitos teóricos bem como as ferramentas utilizadas no presente estudo.

2.1 Abordagem *OBDA* e Ontologias

A abordagem OBDA (Ontology Based Data Access) viabiliza a integração semântica de dados utilizando ontologias e mapeamentos. O paradigma OBDA, também conhecido na literatura como Virtual Knowledge Graph (VKG), surge para lidar com os problemas atuais no desenvolvimento de sistemas de informação, oferecendo uma forma de integração semântica de dados. A ontologia é usada como uma visão conceitual de alto nível dos repositórios de dados, possibilitando que um usuário tenha acesso aos dados sem ter o conhecimento específico de como eles estão organizados em suas fontes e os mapeamentos constroem as relações entre os dados e os termos da ontologia.

Ontologias tem sido uma solução interessante para integração de dados em saúde [4], pois provê um vocabulário controlado de conceitos num domínio, que serve como uma interface conceitual independente do esquema das bases de dados. A utilização de ontologias para acesso a dados distribuídos é proposta no paradigma OBDA [20] como camada de conceitos sobre os dados, oferecendo maior expressividade na descrição do domínio e facilitando o acesso à informação por sistemas computacionais.

A integração de dados é motivada pela necessidade de utilizar dados de múltiplas fontes de maneira efetiva. Frequentemente, esses dados são armazenados de forma heterogênea e contém informações redundantes e inconsistentes. Na abordagem tradicional, a integração de dados envolve atividades de limpeza, deduplicação e homogeneização de dados. Quando se trata de tabelas relacionais, é necessário ainda definir como será representada a informação integrada. Dessa maneira, realizar a integração e desenvolver um mecanismo de acesso conveniente torna-se um processo custoso e intensivo, agravado pela complexidade das bases de dados [20].

Em [19] ontologia é definida como uma especificação explícita e formal de uma conceitualização compartilhada. Uma conceitualização é uma visão de mundo relacionada a um domínio específico, e essa visão é representada por um vocabulário formal, que define um conjunto de conceitos e relações. Ontologias podem ser descritas através da linguagem OWL[3] (Ontology Web Language), parte do conjunto de especificações para a Web Semântica do World Wide Web Consortium (W3C), que inclui também o formato de dados RDF[4] (Resource Description Framework) e a linguagem de consulta SPARQL. A linguagem OWL é projetada para representar conhecimento sobre objetos e suas relações, tendo como base lógica computacional, de modo que ela pode ser utilizada por programas de computador para verificar a consistência de conhecimento ou tornar explícito conhecimento implícito. Ontologias em OWL possuem um vocabulário de classes, propriedades, indivíduos, e dados, e podem ser armazenadas e compartilhadas em documentos RDF.

A ontologia utilizada neste esturdo foi adaptada de OBaS (Ontology for Bariatric Surgery), proposta por [13], foi aplicada no contexto do acompanhamento pós cirurgia bariátrica. O acompanhamento pós-operatório de cirurgia bariátrica é composto por várias etapas, e ocorre durante um período de, no mínimo, dois anos [1]. As ferramentas utilizadas foram o editor de ontologias Protégé[5], e o sistema Ontop.

2.2 Feramentas e Componentes OBDA

A principal característica de sistemas OBDA e VKG é o acesso às fontes de dados apenas quando necessário para uma determinada consulta do usuário, ou seja, ao invés de extrair todos os dados e materializá-los internamente, o grafo de conhecimento é mantido virtual [22]. Uma das principais vantagens dessa característica é que o sistema expõe as informações atualizadas do banco de dados. Dessa forma, o projeto e manutenção dos dados também são simplificados, pois qualquer modificação pode ser testada instantaneamente.

Os componentes principais de um sistema VKG são as consultas do usuário, uma ontologia, mapeamentos e fontes de dados. As recomendação do World Wide Web Consortium (W3C) de linguagens e formatos para esses componentes são, respectivamente: a linguagem SPARQL para consultas, a linguagem OWL2QL para a representação de ontologias, e a linguagem R2RML[6] para os mapeamentos. Diversos sistemas têm sido desenvolvidos para dar suporte a abordagem VKG, dentre eles destacam-se sistemas de reformulação de consultas, ferramentas para o projeto de mapeamento, e ferramentas para a formulação de consultas [21]. Sistemas de reformulação de consultas implementam o núcleo da abordagem VKG, ou seja, a reescrita das consultas em termos da ontologia para consultas sobre as bases de dados relacionais. Exemplos incluem os sistemas: Mastro [5], Morph [7], Ontop, Ultrawrap [17], entre outros. Ferramentas para o projeto de mapeamento são essenciais, já que mapeamentos são o componente central da abordagem VKG. Neste trabalho utilizou-se principalmente as ferramentas Protégé e Ontop.

O Protégé é uma ferramenta de código aberto, desenvolvida na Universidade de Stanford, para criação, manipulação e visualização de ontologias, além de ter uma integração com o Ontop, permitindo a visualização gráfica de funcionalidades deste, como edição dos mapeamentos e execução de consultas *SPARQL*.

O Ontop [7] é uma ferramenta OBDA de código livre, que possibilita a integração semântica de fontes de dados relacionais. O sistema expõe dados heterogêneos como um grafo de conhecimento unificado em RDF gerado a partir do mapeamento dos termos da ontologia com as fontes de dados [6]. Para executar consultas de maneira distribuída, o Ontop utiliza a linguagem SPARQL (Protocol and RDF Query Language)[8]. Por meio dos mapeamentos definidos na ontologia, uma consulta SPARQL é traduzida para SQL (Structured Query Language) e enviada para cada SGBD, os quais retornam o resultado para o Ontop, que apresenta o resultado por completo. Estudos mostram que o desempenho e tempo de execução dessas consultas pode ser comprometido no momento da conversão de SPARQL para SQL. Conforme [4], a complexidade da combinação entre ontologia e mapeamentos implica diretamente no desempenho do Ontop.

[3] https://www.w3.org/TR/owl2-overview/
[4] https://www.w3.org/TR/rdf-primer/
[5] https://protege.stanford.edu/
[6] https://www.w3.org/TR/r2rml/
[7] https://ontop-vkg.org/
[8] https://www.w3.org/TR/sparql11-query/

O desempenho de um sistema OBDA depende criticamente das técnicas de otimização de consultas implementadas. A otimização é essencial para garantir que as consultas SQL geradas pelo sistema possam ser processadas de maneira eficiente pelo SGBD. Uma grande variedade de técnicas de otimização têm sido desenvolvidas, dentre elas, o sistema Ontop utiliza diversas técnicas recentes, como eliminação de joins redundantes utilizando chaves primárias e secundárias; otimização de left joins causados por OPTIONAL e MINUS em consultas SPARQL; e uma nova técnica desenvolvida para a eliminação de self-joins, no caso comum em que os dados estão parcialmente normalizados [22].

2.3 *Linguagens SPARQL e SQL*

Um Sistema Gerenciador de Banco de Dados (SGBD) é um conjunto de programas que dão suporte à criação, utilização e manutenção de banco de dados. Uma das suas principais funções é fornecer respostas dentro de um tempo adequado à consultas de usuários. A utilização de índices pode acelerar o processamento de consultas. Um índice é um conjunto ordenado de valores que contém a chave do índice e ponteiros para os dados da base armazenados em disco [10]. De uma forma geral, a varredura do índice é mais eficiente do que a varredura sequencial da base, pois os dados no índice são preordenados e o seu tamanho geralmente é muito menor que o da base. Logo, ao executar consultas, na maioria das vezes é melhor que o SGBD utilize índices para acessar dados. Na versão do Ontop utilizada (4.1.1), há suporte para seis SGBDs, dentre eles o H2, MySQL e PostgreSQL, utilizados neste estudo.

A estrutura de dados utilizada para representar internamente as consultas no sistema Ontop é a estrutura Intermediate Query (IQ), uma representação uniforme para consultas SPARQL e SQL, que tem como base álgebra relacional. Essa estrutura foi desenvolvida para atender as funcionalidades introduzidas pela versão 1.1 da linguagem SPARQL, dentre elas a funcionalidade de agregação [22]. A transformação das consultas SPARQL para SQL envolve a reescrita e desdobramento, que é realizada utilizando a estrutura IQ. Quando a transformação é finalizada, a expressão em IQ é convertida para SQL e executada pelo sistema de gerenciamento de banco de dados relacional. Quase todo o processamento de consultas é delegado ao sistema de gerenciamento de banco de dados, e o Ontop realiza apenas a projeção dos resultados da consulta para termos RDF.

Diferente de SPARQL, que tem sua sintaxe e semântica padronizada, a linguagem SQL é mais variada devido a diferentes sistemas de gerenciamento de bancos de dados que não seguem estritamente os padrões ANSI/ISO. Por esse motivo, é difícil garantir a geração de consultas SQL que sejam compatíveis com múltiplos sistemas. Dado a diversidade do ecossistema SQL, o Ontop modela dialetos SQL de maneira granular, considerando seus tipos de dados, convenções em termos de atributos, tabelas, e modificadores de consultas, semântica de funções, restrições em cláusulas, e estrutura do catálogo de dados [22]. Além disso, o Ontop permite utilizar funções SQL arbitrárias, inclusive definidas pelo usuário, nas consultas do mapeamento.

A versão 4 do sistema Ontop é compatível com quase todas as características do padrão SPARQL 1.1. Entre as funcionalidades ainda não implementadas, é possível citar *property paths*, `[NOT] EXISTS`, algumas funções SPARQL sem tradução direta para SQL, e funções *hash*, *replace* e *regex* que tem suporte limitado pois dependem da implementação do SGBD. Em [12] foram avaliados os sistemas OBDA Ontop e Mastro nos benchmarks NPD e ACI, ambos com ontologias complexas. Em geral, Ontop foi mais rápido em NPD, enquanto Mastro foi mais rápido em ACI. Essa e outras avaliações confirmam que, embora o sistema Ontop não seja sempre o mais rápido, sua performance é bem robusta. Ao escolher um sistema VKG, performance é apenas uma das dimensões a serem consideradas: nas perspectivas de usabilidade, completude, e solidez, o sistema Ontop se destaca entre os sistemas de código aberto disponíveis [22].

Além de dar suporte a mapeamentos R2RML, o Ontop também possui sua própria linguagem para mapeamentos, que é intuitiva e fácil de aprender e usar [6]. Um mapeamento na linguagem nativa do Ontop consiste de uma fonte (*source*), que é a consulta SQL que obtêm valores do banco de dados, e um alvo (*target*), que é um template utilizado para construir triplas RDF com os valores obtidos. O sistema também inclui ferramentas para converter mapeamentos nativos em mapeamentos R2RML, e vice-versa.

2.4 Dados Abertos em Saúde

Dados abertos são definidos como dados que podem ser livremente usados, modificados e compartilhados por qualquer pessoa para qualquer propósito - restrito, no máximo, a medidas que preservam a proveniência e abertura destes dados [15].

Na área da saúde, o Departamento de Informática do Sistema Único de Saúde (DATASUS) foi criado em 1991, com o propósito de prover os órgãos do SUS de sistemas de informação e suporte de informática. Estes sistemas contêm dados coletados referentes ao SUS e têm como objetivo auxiliar no processo de planejamento e tomada de decisões [3]. Os sistemas de informação são utilizados nas diversas áreas da saúde pública. Dentre os disponíveis para a população, podem conter conjuntos de dados sobre estatísticas vitais, como o Sistema de Informações de Nascidos Vivos (SINASC) e Mortalidade (SIM); sobre epidemiologia, como o Sistema de Agravos de Notificação Compulsória (SINAN); e sobre a rede de assistência à saúde, como o Sistema de Informações Ambulatoriais (SIASUS) e Hospitalares (SIHSUS) do SUS, entre outras áreas.

No trabalho de Silva e Autran [18], é concluído que o Datasus realiza a distribuição dos dados cumprindo com os 8 princípios dos dados abertos governamentais. A disseminação de informações sobre a saúde pública também é parte das competências do Datasus, e é realizada com o auxílio dos sistemas Tabnet e TabWin; este segundo será discutido posteriormente neste relatório.

Dados pertinentes à cirurgia bariátrica na rede pública estão presentes nos sistemas SIASUS e SIHSUS. Os arquivos contendo estes dados podem ser adquiridos gratuitamente no site do Datasus, e são divididos pelo mês e estado em que as informações foram divulgadas. O SIHSUS contém os registros de AIH (Autorização de Internação Hospitalar) para todos os atendimentos provenientes de internações hospitalares financiados pelo SUS. O SIASUS possui informações sobre a produção ambulatorial através de laudos de APAC (Autorização de Procedimento de Alta Complexidade), que são separados de acordo com o tipo de atendimento, dentre eles incluso o acompanhamento a cirurgia bariátrica.

2.5 Trabalhos Correlatos

Diversos trabalhos de integração de dados têm sido realizados através do sistema Ontop. Em [14], o Ontop é utilizado para realizar consultas SPARQL sobre os acórdãos do STF, reutilizando uma ontologia do domínio jurídico. Os dados dos acórdãos foram obtidos em um banco de dados NoSQL MongoDB e adaptados para um banco de dados relacional. Para classificar as decisões dos acórdãos em favoráveis e não favoráveis, foi aplicada uma técnica de machine learning.

Em [9] é apresentado o SemanticSUS, um portal semântico para acesso, análise e visualização de dados do SUS. O portal publica duas bases de dados do SUS, SIM (Sistema de Informação sobre Mortalidade) e SINASC (Sistema de Informação sobre Nascidos Vivos).

Em [16] é descrito o processo de construção incremental do grafo de conhecimento do SemanticSUS, utilizando a abordagem OBDA. Foi utilizada uma técnica de *bootstrapping* para geração automática de ontologias através do Ontop, em que cada ontologia exportada é tratada como um vocabulário parcial do domínio.

3 PROCEDIMENTOS METODOLÓGICOS

Esta seção apresenta os procedimentos desenvolvidos e ferramentas utilizadas para realizar os experimentos, conforme ilustra a figura 1. Primeiro obteve-se os dados abertos disponíveis no portal do DATASUS, que se encontravam em formato proprietário *.dbc*. Para conversão foi utilizada a ferramenta TabWin, disponibilizada pelo DATASUS que permite conversão *.dbc* para *.dbf* (Um outro formato proprietário do DATASUS) e então exportar em *.csv*. Assim todos os dados puderam ser armazenados nos bancos de dados estudados (MySQL, H2 e PostgreSQL).

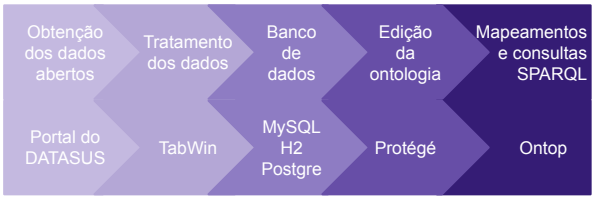

Figure 1: Etapas e Ferramentas utilizadas.

Na camada ontológica, utilizou-se uma extensão da ontologia denominada OBaS (*Ontology for Bariatric Surgery*) [13]. A ontologia foi editada utilizando o *Protégé*. Os mapeamentos e consultas SPARQL foram implementados no plugin *Ontop*, com a ferramenta *Protégé*.

3.1 Bases utilizadas

Para o estudo foram utilizadas duas bases de dados abertas sobre cirurgia bariátrica do Sistema Único de Saúde Brasileiro (SUS): 1) ABOPR, que contém dados do Sistema de Informações Ambulatoriais do SUS (SIASUS) sobre laudos de Autorização de Procedimento de Alta Complexidade (APAC) de acompanhamento a cirurgia bariátrica e 2) RDPR, que possui dados do Sistema de Informações Hospitalares do SUS (SIHSUS) sobre registros de Autorização de Internação Hospitalar (AIH) de todas as internações hospitalares custeadas pelo SUS. Ambas as bases possuem dados referentes a procedimentos realizados no estado do Paraná.

A partir das duas bases foram geradas quatro cargas de dados, sendo elas:

- B1 com dados de 2014 e 2015 e com 14,2 MB (megabytes) na ABOPR e 678 MB na RDPR;
- B2 com dados de 2014 a 2017 e com 38,2 MB na ABOPR e 1.438 MB na RDPR;
- B3 com dados de 2014 a 2019 e com 72,2 MB na ABOPR e 2.255 MB na RDPR; e,
- B4 com dados de 2014 a 2021 e com 84,8 MB na ABOPR e 2.957 MB na RDPR.

3.2 SGBDs e Consultas utilizadas

Para o estudo foram selecionados os seguintes SGBDs relacionais H2[9] (versão 1.4.196), MySQL[10](versão 8.0.29) e o PostgreSQL[11] (versão 14.3). Os três foram selecionados por serem de distribuição livre e por terem suporte para o Ontop. Todos os SGBDs foram utilizados com a configuração padrão. Em um estudo preliminar [2] utilizando o Ontop e o H2 pode-se observar que o tempo de resposta foi insatisfatório, motivando um aprofundamento com foco no desempenho de consultas com mais SGBDs.

As consultas foram elaboradas para responder: Q1) “As comorbidades dos pacientes implicam no custo da cirurgia bariátrica?" e Q2) “A prática de atividades físicas e alimentação saudável pelos pacientes implica no custo da cirurgia bariátrica?”. Ambas consultas envolvem a integração entre as duas bases do experimento.

A figura 2 mostra o *script* SPARQL da consulta Q1. AS três primeiras linhas do *script* define o prefixo da ontologia. A consulta consiste das linhas 4–8. A linha 4 define os atributos que serão retornados pela consulta. Para cada internação, recupera-se a pessoa, as comorbidade da pessoa e o valor da internação. As linhas 6–8 definem o grafo que será recuperado das bases de dados. Uma internação que tenha relação com pessoa (linha 6), o valor dessa internação (linha 7) e as comorbidades da pessoa internada, caso exista (linha 8). A Figura 3 mostra o *script* da consulta Q2. Aqui, no lugar das comorbidades, os hábitos das pessoas são considerados. Assim, recupera-se (caso exista) hábitos que a pessoa tenha adotado como, por exemplo, a prática de alimentação saudável ou de exercício físico.

Os *scripts* SQL (gerado pelo *Ontop*) com as consultas estão disponíveis em https://bit.ly/3JlGdrI.

4 EXPERIMENTOS E RESULTADOS

Essa seção apresenta os experimentos práticos e seus resultados. Para explorar o desempenho das consultas, duas formas de indexação foram utilizadas em cada SGBD, sendo elas: I0 - sem utilização de índices; e I1 - indexação dos atributos envolvidos na operação de integração de dados entre RDPR e ABOPR, além da indexação dos atributos envolvidos no acesso aos dados da ABOPR, relacionados à comorbidade em Q1 e à prática de atividades físicas e alimentação

[9] http://h2database.com/
[10] https://www.mysql.com/
[11] https://www.postgresql.org/

```
1 PREFIX :
2 <https://github.com/glaubernunes/ontology-for-bariatric
3 -surgery/raw/main/ontology-for-bariatric-surgery.owl#>
4 SELECT ?internacao ?pessoa ?comorbidade ?valor
5 WHERE {
6 ?internacao :temPessoa ?pessoa .
7 ?internacao :custo ?valor .
8 OPTIONAL {?pessoa :OBAS_0000172 ?comorbidade} }
```

Figure 2: Script *SPARQL* que recupera para cada internação a pessoa, o custo e a comorbidade (caso existam).

```
1 PREFIX :
2 <https://github.com/glaubernunes/ontology-for-bariatric
3 -surgery/raw/main/ontology-for-bariatric-surgery.owl#>
4 SELECT ?internacao ?pessoa ?habito ?valor
5 WHERE {
6 ?internacao :temPessoa ?pessoa .
7 ?internacao :custo ?valor .
8 OPTIONAL {?pessoa :OBAS_0000173 ?habito} }
```

Figure 3: Script *SPARQL* que recupera para cada internação a pessoa, o custo e se adotou alimentação saudável ou prática de exercício físico (caso tenham adotado).

saudável em Q2. A indexação de dados é um recurso que pode minimizar o tempo de processamento de consultas reduzindo o volume de dados acessados pelas mesmas.

De acordo com [4] o custo de responder uma consulta com o *Ontop* pode ser dividido em três partes: *i)* o custo de gerar a consulta SQL, *ii)* o custo de execução do SGBD e *iii)* o custo de transformar os resultados SQL em termos RDF. Portanto, o experimento de desempenho das consultas envolveu duas análises, a saber: *i)* o tempo de processamento do *Ontop* na execução das consultas, a fim de responder a pergunta: *Algum SGBD pode minimizar a carga de processamento do Ontop e, consequentemente, o seu tempo de processamento?*; e, *ii)* uma análise isolada dos SGBDs, a fim de responder a pergunta: *Qual SGBD pode minimizar o tempo de integração de dados com o Ontop?*.

Quanto aos equipamentos utilizados, o experimento foi realizado em um Dell G3 3590, Intel Core i5-9300H (CPU 2.40GHz) com 8 GB de RAM. Para medir o tempo de resposta, utilizou-se o contador das próprias ferramentas utilizadas. Cada experimento foi executado dez vezes e considerou-se o tempo médio em segundos obtido.

A Tabela 1 apresenta os tempos de execução das consultas Q1 e Q2 (coluna Q) nas bases (coluna B), combinando o *Ontop* com cada SGBD estudado, usando índice (I1) ou não (I0). O tempo de execução no H2 não foi considerado para I0 dada a demora do sistema em responder as consultas. É possível notar na Tabela 1 que o PostgreSQL apresentou melhor desempenho que os demais SGBDs tanto com I0 quanto com I1.

Em um segundo experimento, os *scripts* SQL gerados pelo *Ontop* foram executados em cada SGBD. Um mesmo *script* SQL foi gerado pelo *Ontop* para cada consulta, independentemente do SGBD e da indexação adotada pelo mesmo. A Tabela 2 exibe os tempos obtidos desconsiderando-se o tempo de processamento do *Ontop* nas consultas. A diferença dos valores das Tabela 1 (valores com *Ontop*) e Tabela 2 (valores sem o *Ontop*) viabilizou a análise da carga de processamento do *Ontop*. A Tabela 3 mostra o tempo de processamento do *Ontop* com I1. Para cada consulta e cada base, o maior tempo de processamento exigido do *Ontop* foi representado com um asterisco (*) na Tabela 3.

É possível notar na Tabela 3 que o tempo de processamento do *Ontop* aumentou conforme o tamanho da base aumentou, dado que o número de respostas das consultas também aumentaram. Analisando cada SGBD, o H2 exigiu um maior tempo do *Ontop* quando as consultas envolveram a maior base do experimento, a B4. Já o MySQL implicou um maior tempo de processamento no *Ontop* durante a execução de Q1 e o PostgreSQL durante a execução de Q2. Logo, nenhum SGBD se destacou no quesito de minimizar a carga de processamento do *Ontop*.

É importante notar que em alguns casos o tempo de processamento do *Ontop* foi maior que o tempo de processamento do SGBD. Por exemplo, enquanto o MySQL levou 0.5 segundos para executar Q1 em B1 (Tabela 2) o *Ontop* levou 1.7 segundos (Tabela 3). O mesmo ocorreu com o PostgreSQL em Q2-B3. A variação dos tempos do *Ontop* levantada no experimento pode estar relacionada ao custo de transformação dos resultados SQL para termos RDF, o qual depende da complexidade da combinação da ontologia com os mapeamentos e não necessariamente do tamanho da base, como já foi observado por [4].

		I0			I1		
Q	**B**	**H2**	**MySQL**	**PostgreSQL**	**H2**	**MySQL**	**PostgreSQL**
Q1	**B1**	-	6.2	1.3	6.6	2.2	1.3
	B2	-	9.7	6.3	15.1	5.7	2.7
	B3	-	17.1	8.9	55.8	12.9	3.7
	B4	-	19.7	12.2	84.9	16.4	4.1
Q2	**B1**	-	6.1	1.6	6.2	1.1	1.7
	B2	-	11.1	6.9	13.8	6.4	4.8
	B3	-	23.1	11	51.6	14.7	6.7
	B4	-	26.7	15.3	92.2	19.2	7.8

Table 1: Tempo médio (em segundos) de Q1 e Q2 envolvendo o Ontop e os SGBDs com I0 e I1

		I0			I1		
Q	**B**	**H2**	**MySQL**	**PostgreSQL**	**H2**	**MySQL**	**PostgreSQL**
Q1	**B1**	-	5.2	0.5	6.3	0.5	0.8
	B2	-	9.1	4.5	14.2	4.1	1.8
	B3	-	16.2	6.8	53.8	10.2	2.2
	B4	-	19.1	8.1	80.3	13.9	2.3
Q2	**B1**	-	4.5	0.7	5.8	0.7	0.9
	B2	-	10.1	5.6	13	5.3	2.6
	B3	-	20.7	7.8	49	12.3	2.9
	B4	-	25.9	14.3	85.6	17.1	3.3

Table 2: Tempo médio (em segundos) de Q1 e Q2 nos SGBDs sem o Ontop

A análise isolada dos SGBDS foi elaborada com base nos planos de execução das consultas Q1 e Q2 obtidos pelo comando *explain analyze* nos SGBDs. O H2 não foi considerado nessa seção por não disponibilizar esse comando aos seus usuários. Os experimentos consideraram a indexação padrão do MySQL e do PostgreSQL,

Q	B	H2	MySQL	PostgreSQL
Q1	B1	0.3	*1.7	0.5
	B2	0.9	*1.6	0.9
	B3	2	*2.7	1.5
	B4	*4.6	2.5	1.8
Q2	B1	0.4	0.4	*0.8
	B2	0.8	1.1	*2.2
	B3	2.6	2.4	*3.8
	B4	*6.6	2.1	4.5

Table 3: Tempo médio (em segundos) de execução do Ontop com a indexação I1

sendo que ambos adotam a mesma estrutura de dados na criação de índices, a Árvore-B. Por fim, a escalabilidade de dados dos SGBDs também foi analisada, que é a capacidade do SGBD manter o seu desempenho conforme aumenta a carga de dados processada.

O gráfico da Figura 4 mostra o tempo de processamento do *script* SQL (gerado pelo *Ontop*) de Q1 nos SGBDs, indexados ou não, em B1–B4 (conforme Tabela 2). A consulta Q2 apresentou o mesmo padrão que Q1. Observa-se no gráfico que a indexação foi essencial para os dois SGBDs analisados, uma vez que o desempenho deles com indexação (I1) foi superior em relação ao processamento sem indexação (I0).

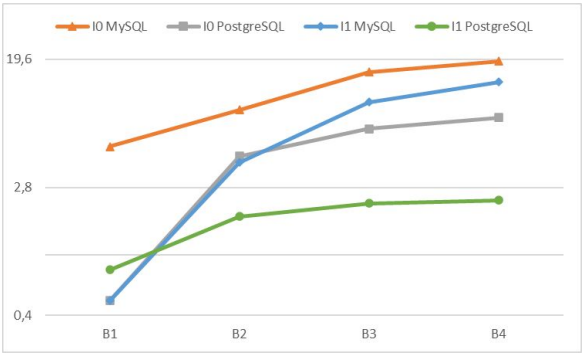

Figure 4: Tempo médio (em segundos) de execução de Q1 nos SGBDs (Gráfico em escala logarítmica).

Dada uma consulta, o otimizador de cada SGBD adota uma estratégia para a definição do seu plano de execução.

O otimizador leva em conta, por exemplo, o número de operações I/O do disco (acesso à blocos do disco) e o custo de processamento da CPU. Por fim, o otimizador escolhe e ordena um conjunto de operações gerando o plano de execução da consulta. O objetivo do otimizador é gerar um plano de execução que tenha um bom desempenho. Além de outras operações, a execução de Q1 e Q2 envolveu, respectivamente, o acesso aos dados relacionados à comorbidade e hábitos saudáveis em ABOPR; e, a integração de dados das bases ABOPR e da RDPR.

Conforme o gráfico da figura 4, a estratégia de planejamento de consultas do PostgreSQL apresentou melhores resultados do que a do MySQL.

A operação de acesso aos dados pode explorar ou não a indexação. A escolha depende da estratégia do otimizador de cada SGBD. A operação *Full Table Scan*, que lê todo o conteúdo de uma base de forma sequencial, foi adotada na maioria dos planos de execução do experimento. O PostgreSQL optou pelo *Full Table Scan* em todos os seus planos. Já, o MySQL adotou o *Full Table Scan* somente com I0. Com I1, o MySQL explorou índices adotando operações tais como, por exemplo, *Index Range Scan* e *Full Index Scan*. Para a integração dos dados das duas bases, ambos os sistemas seguiram o mesmo padrão na elaboração dos planos de execução adotando o *Hash Join* para integrar os dados das bases não-indexadas e o *Nested Loop Join* nas bases indexadas. Quanto à escalabilidade de dados, observa-se no gráfico da figura 4 (com escala logarítmica) que o PostgreSQL apresentou uma taxa de crescimento de tempo menor que o MySQL conforme o volume de dados processado por Q1 aumentou de B1 até B4.

5 CONSIDERAÇÕES FINAIS

Esta pesquisa buscou verificar o desempenho de consultas utilizando Ontop e SGBDs relacionais. O experimento envolveu análise sobre o tempo de processamento do Ontop na execução das consultas, buscando-se saber qual dos SGBDs utilizados no estudo podem minimizar a carga de processamento do Ontop e, consequentemente, o seu tempo de processamento. Também buscou-se saber qual SGBD pode minimizar o tempo de integração de dados com o Ontop.

Esta pesquisa pode contribuir com desenvolvedores de SIS na escolha do SGBD a ser utilizado em projetos de integração que utilizam ferramentas OBDA. Conforme exposto, verificou-se que entre os três, o PostgreSQL teve o menor tempo de resposta na execução das duas consultas, portanto o melhor desempenho. Sugere-se também que desenvolvedores com conhecimento básico sobre a configuração de SGBDs utilizem o PostgreSQL, pois neste estudo foram utilizadas a configuração padrão. Em segundo sugere-se o uso do MySQL e o H2 como terceira opção (dada as configurações utilizadas).

Nas próximas etapas da pesquisa pretende-se aprofundar o estudo, explorando outros tipos de consultas, indexação, carga de dados e mapeamentos. Vislumbra-se um estudo sobre o custo de processamento referente ao *Ontop* a fim de propor alternativas que melhorem o desempenho do processo de integração de dados. O estudo considerará especificamente o processo de desserialização dos dados recebidos dos SGBDs pelo *Ontop*. Cada SGBD possui o seu JDBC (*Java Database Connectivity*) responsável pela sua comunicação com o *Ontop*. Logo, o estudo analisará cada JDBC a fim de detectar as suas características quando relacionado com o *Ontop*.

ACKNOWLEDGMENTS

Ao Programa Institucional de Bolsas de Iniciação Científica (PIBIC/CNPq-FA-UEM) pela bolsa concedida ao segundo autor e ao PIC-UEM que possibilitou a participação do primeiro autor na pesquisa.

REFERENCES

[1] ABESO. 2016. Diretrizes brasileiras de obesidade. https://abeso.org.br/wp-content/uploads/2019/12/Diretrizes-Download-Diretrizes-Brasileiras-de-Obesidade-2016.pdf. Acessado em: 15-12-2021. (2016).

[2] Samuel Bispo, Vinícius Fukace, Ana Mazur, Raqueline Penteado, and Heloise Teixeira. 2022. Integração de dados abertos em saúde com o modelo obda: um estudo de caso na área de cirurgia bariátrica. In *Anais do XXII Simpósio Brasileiro de Computação Aplicada à Saúde*. SBC, Teresina, 401–412. DOI: 10.5753/sbcas.2022.222720.

[3] Brasil. 2020. Datasus: Histórico. Ministério da Saúde, Secretaria Executiva, Departamento de Informática do SUS. (2020). Retrieved Oct. 24, 2021 from https://datasus.saude.gov.br/sobre-o-datasus/.

[4] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. 2017. Ontop: answering SPARQL queries over relational databases. *Semantic Web*, 8, 3, 471–487. DOI: 10.3233/SW-160217.

[5] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, and D. F. Savo. 2011. The mastro system for ontology-based data access. *Semantic Web*, 2, 1, 43–53.

[6] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. 2016. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8, 3, (Dec. 2016), 471–487. Óscar Corcho, (Ed.) DOI: 10.3233/SW-160217.

[7] David Chaves-Fraga, Luis Pozo, Jhon Toledo, Edna Ruckhaus, and Oscar Corcho. 2020. Morph-CSV: Virtual Knowledge Graph Access for Tabular Data. In *ISWC 2020 Demos*. (Oct. 2020).

[8] Giliate Cardoso Coelho Neto and Arthur Chioro. 2021. Afinal, quantos sistemas de informação em saúde de base nacional existem no brasil? *Cadernos de Saúde Pública*, 37, Cad. Saúde Pública, 2021 37(7). DOI: 10.1590/0102-311X00182119.

[9] Matheus Mayron Lima Da Cruz, Caio Viktor Silva Avila, Vânia Maria Ponte Vidal, and Narciso Moura Arruda Junior. 2019. SemanticSUS: Um Portal Semântico baseado em Ontologias e Dados Interligados para Acesso, Integração e Visualização de Dados do SUS. *Anais Estendidos do Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS 2019)*, (June 2019), 13–18. DOI: 10.5753/sbcas.2019.6277.

[10] M.V. Mannino. 2008. *Projeto, Desenvolvimento de Aplicações e Administração de Banco de Dados - 3.ed.* AMGH Editora. ISBN: 9788580553635. https://books.google.com.br/books?id=Pjq6AwAAQBAJ.

[11] Jose Mora and Oscar Corcho. 2013. Engineering optimisations in query rewriting for obda. In *Proceedings of the 9th International Conference on Semantic Systems* (I-SEMANTICS '13). Association for Computing Machinery, Graz, Austria, 41–48. ISBN: 9781450319720. DOI: 10.1145/2506182.2506188.

[12] Manuel Namici and Giuseppe De Giacomo. 2018. Comparing Query Answering in OBDA Tools over W3C-Compliant Specifications. en. *31st International Workshop on Description Logics*.

[13] Glauber Muzyka Oyarzabal Nunes. 2021. *MultiOnto: Método de Construção de Ontologia Considerando Heterogeneidade de Fontes e Tipos de Conhecimentos - Um Estudo de Caso sobre Cirurgia Bariátrica*. Master's thesis. Universidade Tecnológica Federal do Paraná.

[14] Rafael Brito de Oliveira. 2017. *Utilização de ontologias para busca em base de dados de acórdãos do STF*. pt. Mestrado em Ciência da Computação. Universidade de São Paulo, São Paulo. DOI: 10.11606/D.45.2018.tde-24012018-110738.

[15] OPEN KNOWLEDGE FOUNDATION. 2009. The Open Definition: Defining Open in Open Data, Open Content and Open Knowledge. (2009). Retrieved Oct. 15, 2021 from https://opendefinition.org/.

[16] Tulio Rolim, Caio Avila, Narciso Arruda, José Silva, José Maia, Mauro Oliveira, Luiz Andrade, and Vânia Vidal. 2020. Um Enfoque Incremental para Construção do Grafo de Conhecimento do SUS. *Anais Principais do Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS 2020)*, (Sept. 2020), 72–83. DOI: 10.5753/sbcas.2020.11503.

[17] Juan Sequeda and Daniel Miranker. 2013. Ultrawrap: SPARQL execution on relational data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 22, (Oct. 2013), 19–39. DOI: 10.1016/j.websem.2013.08.002.

[18] Pollianna Marys de Souza e Silva and Marynice Medeiros Matos de Autran. 2019. Repositório DATASUS: Organização e Relevância dos Dados Abertos em Saúde para a Vigilância Epidemiológica. *P2P E INOVAÇÃO*, 6, (Oct. 2019), 50–59. DOI: 10.21721/p2p.2019v6n1.p50-59.

[19] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. 1998. Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25, 1, 161–197. DOI: https://doi.org/10.1016/S0169-023X(97)00056-6.

[20] G. Xiao, D. Hovland, D. Bilidas, M. Rezk, M. Giese, and D. Calvanese. 2018. Efficient ontology-based data integration with canonical iris. In *European Semantic Web Conference*. Springer, 697–713.

[21] Guohui Xiao, Linfang Ding, Benjamin Cogrel, and Diego Calvanese. 2019. Virtual Knowledge Graphs: An Overview of Systems and Use Cases. *Data Intelligence*, 1, (May 2019), 201–223. DOI: 10.1162/dint_a_00011.

[22] Guohui Xiao et al. 2020. The Virtual Knowledge Graph System Ontop. en. In *The Semantic Web – ISWC 2020*. Vol. 12507. Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, (Eds.) Series Title: Lecture Notes in Computer Science. Springer International Publishing, Cham, 259–277. ISBN: 978-3-030-62466-8. DOI: 10.1007/978-3-030-62466-8_17.