

Os desafios de uma aplicação de *Carpooling* no contexto de uma comunidade universitária brasileira

Tiago Stapenhorst Martins¹, Nádia P. Kozievitch¹

¹Dep. de Informática, Universidade Tecnológica Federal do Paraná, Curitiba, PR, Brasil

tiagosmx@gmail.com, nadiap@utfpr.edu.br

Abstract. *Urban transport is intrinsically linked to the operation of large cities with dense traffic as an important factor in the generation of large jams, and lack of parking, among others. One solution to this problem is to share a ride (also called carpooling or ride sharing). The principle of carpooling explores the unused space available in private transport populating them with people or items that have shared destinations. This study explores the challenges regarding: (i) the concept of carpooling; (ii) the problem of applying an algorithm to a road network; and (iii) other issues regarding GIS applications, considering a university community in Brazil.*

Resumo. *O transporte urbano está intrinsecamente ligado ao funcionamento de grandes metrópoles, tendo o tráfego denso como fator importante na geração de grandes congestionamentos, falta de estacionamentos, entre outros. Uma possível solução para este problema é o compartilhamento de carona (também chamado de carona solidária ou agendamento de caronas) ou carpooling. O princípio do carpooling é aproveitar o espaço não usado mas disponível em meios de transporte privados populando-os com pessoas ou itens que têm destinos em comum. Este trabalho apresenta desafios considerando: (i) o conceito de carpooling; (ii) o problema de aplicar algoritmos a uma rede viária; e (iii) outras questões relacionadas a aplicações GIS, considerando uma comunidade universitária brasileira.*

1. Introdução

Ao andar pelas ruas de centros urbanos é possível perceber que ao longo dos anos houve um aumento na quantidade de carros e ônibus em circulação, especialmente nas regiões centrais onde o tráfego de pessoas costuma ser maior principalmente nos horários de pico. Aparentes políticas econômicas e de urbanismo de incentivo do uso de carros, fortemente enraizadas na cultura estado-unidense, favorecem para o aumento da população de automóveis[Miller et al. 1999]. Basta uma observação simples do interior dos carros que trafegam nas ruas - dificilmente vê-se nos horários de pico um carro com todos os assentos ocupados enquanto usuários de ônibus experimentam uma superlotação¹. Além disso, o transporte rodoviário é responsável por aproximadamente 16% do CO₂ produzido por seres humanos².

¹<http://www1.folha.uol.com.br/fsp/cotidian/ff0404201014.htm> Acesso em: 18/01/2014

²<http://oica.net/> Acesso em 01/12/2014.

Cidades brasileiras como São Paulo já implantaram um sistema de rodízio de carros³, em que determinados dias da semana, apenas placas iniciadas com determinadas letras podem transitar. Em grandes centros urbanos como Curitiba, problemas com a mobilidade urbana estão presentes. Carros, ônibus, motocicletas e bicicletas representam alguns dos meios de transporte mais comuns utilizados na cidade. Ainda que 45% da população utilize o transporte coletivo⁴ (caracterizado pelos ônibus públicos), Curitiba ficou em primeiro lugar (em comparação a todas as capitais brasileiras) em proporção de carro por habitante⁵. Em regiões de tráfego intenso (como o centro de Curitiba), onde (i) edificações não possuem estacionamento próprio (como a UTFPR), e (ii) estacionamentos particulares praticam um custo elevado, a mobilidade cada vez mais tem se tornado um desafio.

Em particular, na cidade de Curitiba, há uma preocupação dos usuários de ônibus com relação à sua segurança física - os assentos dos ônibus não têm cintos de segurança e muitas pessoas passam viagens inteiras de pé, o que aumenta a chance de queda e de riscos à saúde a algumas pessoas. As reclamações sobre a comodidade e segurança do transporte público e a redução do número de carros são preocupações presentes no espaço das grandes cidades⁶. Vendo diariamente que carros trafegam pelas ruas transportando às vezes uma ou duas pessoas desafia o sentido da eficiência do transporte urbano.

Suponha, por exemplo, que na Figura 1-A a seta indicada por A ilustra a localização da Universidade Tecnológica Federal do Paraná (UTFPR) e as outras, a localização de pessoas. Dado que todas elas precisam estar na UTFPR no mesmo horário e que E e H têm carro e seus trajetos são representados pelo caminho de A até B percebe-se que C e D são candidatos a pegar carona com E pela proximidade, o mesmo acontece com I, K e F para o motorista H. Os motoristas não precisam necessariamente pegar todas as pessoas, ficando a critério dele o raio de distância que estará disposto a sair de sua rota tradicional para dar carona a alguém e ao número de assentos disponíveis.

Deve-se levar em conta que o trajeto original versus o trajeto com caroneiros não deve extrapolar um limite aceitável para o motorista e os trajetos devem levar em conta o menor caminho entre os pontos, o que pode auxiliar numa economia de tempo e combustível. Dentro do limite estipulado pelo motorista são considerados: o número de vagas disponíveis no carro, a proximidade de um caroneiro com um raio de x quilômetros a partir do trajeto original do motorista e que o trajeto novo não ultrapasse uma porcentagem 'x' a mais do que o caminho original (o valor de x é escolhido pelo motorista).

A Figura 1-B contempla outro caso, onde os trajetos de A até B mostram o caminho que as pessoas B e C fazem para chegar à UTFPR. A linha mais escura mostra uma possibilidade de caminho que permita C dar carona à A. Este exemplo ilustra os possíveis casos em que implicarão na redução do número de carros no trânsito.

³<http://www.cetsp.com.br/consultas/rodizio-municipal/como-funciona.aspx> Acesso em: 18/01/2014

⁴<http://www.biocidade.curitiba.pr.gov.br/biocity/33.html> Acesso em 02/06/2014.

⁵<http://exame.abril.com.br/brasil/noticias/curitiba-e-capital-com-mais-carros-por-pessoa-veja-ranking> Acesso em 02/06/2014.

⁶<http://www.gazetadopovo.com.br/opiniao/conteudo.phtml?id=1355589> Acesso em: 18/01/2014

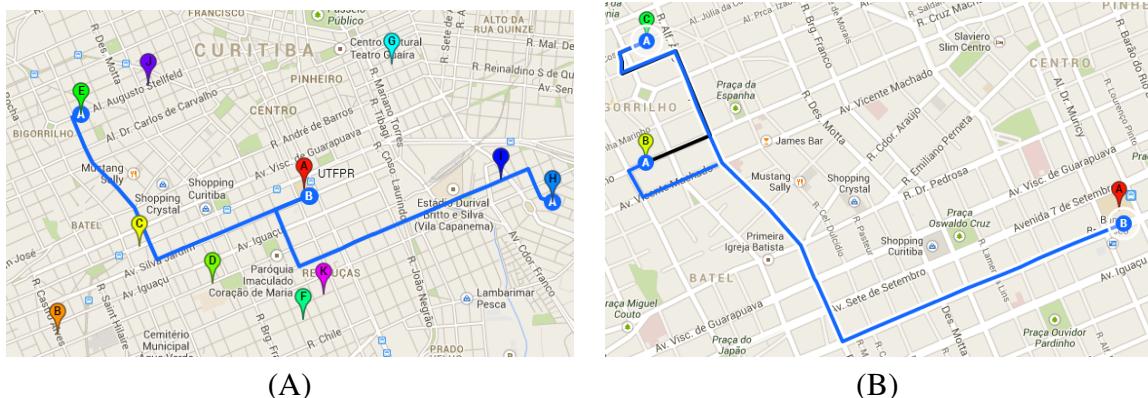


Figura 1. Mapa com a localização de residências e a UTFPR: (A) Trajeto 1 e (B) Trajeto 2.

Uma possível solução para este problema é o compartilhamento de carona (também chamado de carona solidária ou agendamento de caronas) ou *carpool*. O princípio do *carpool* é aproveitar o espaço não usado mas disponível em meios de transporte privados populando-os com pessoas ou itens que têm destinos em comum [Chen et al. 2011, Megalingam et al. 2011, Sghaier et al. 2011, Yan et al. 2011] numa forma de socialização do transporte particular. O uso compartilhado de um carro único por 2 ou mais pessoas pode reduzir o número de carros nas ruas. *Carpool* ajuda o ambiente, permitindo a utilização de combustível de modo inteligente, na redução de poluição e problemas de saúde. Ele reduz o tráfego - e consequentemente - o tempo que as pessoas gastam em seus carros. O *Carpool* também tem o potencial de impactar aspectos sociais, já que permite a maior interação entre pessoas [Graziotin 2013].

A eficiência da mobilidade viária poderia ser maior, com carros levando pessoas com trajetos diários semelhantes. Este trabalho apresenta os conceitos, as aplicações, um protótipo e os desafios relacionados a *carpool*, dentro do contexto de uma comunidade universitária brasileira. A proposta, apesar de envolver a questão de melhorar o fluxo do transporte, não envolve grandes investimentos em infraestrutura, construção de pontes ou vias. Além disso, serve como forma de prevenção para que políticas públicas drásticas (como o rodízio em São Paulo) ocorram em menor quantidade. O restante do artigo está disposto entre diferentes seções: a seção 2 apresenta os trabalhos relacionados; a seção 3 apresenta a visão geral da aplicação proposta, e a seção 4 apresenta a conclusão e trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção são apresentados os trabalhos relacionados referentes a Geoprocessamento e *carpooling*.

2.1. Geoprocessamento

Geoprocessamento corresponde a um conjunto de conhecimentos matemáticos e tecnologias computacionais para tratamento e criação de informações georreferenciadas, baseando-se e influenciando as áreas como cartografia, análise de recursos naturais, transporte, comunicações, energia e planejamento urbano [Rocha 2002, Braga et al. 2008, Hamada and do Valle Gonçalves 2007].

O geoprocessamento vem sendo utilizado para fins comerciais, acadêmicos e governamentais sempre visando a integração de dados espaciais e não espaciais em seus projetos [Hamada and do Valle Gonçalves 2007]. Alguns exemplos de suas aplicações são:

- Manejo e conservação de recursos naturais: Sob forma de estudos de impacto ambiental, modelagem de águas subterrâneas e caminhamento dos contaminantes, estudos de migrações e habitats das faunas, pesquisas de potencial mineral, etc.
- Gestão das explorações agrícolas: Envolvendo a gestão de cultivo do campo, manejo de irrigação, avaliação do potencial agrícola, etc.
- Planejamento de área urbana: Abordando o planejamento dos transportes, desenvolvimento de planos de evacuação, localização de acidentes, seleção de itinerários de viagens, mobilidade urbana [Sassi et al. 2014].
- Gestão das instalações: Avaliando a localização de cabos e tubulações para o planejamento e manutenção das instalações.
- Administração pública: Servindo para gestão de cadastros, avaliação predial/territorial, gestão da qualidade das águas, planos de organização, etc.
- Comércio: Abordando análises de estruturas de mercado, planejamento de desenvolvimento, análises de concorrência e das tendências de mercado entre outros.

Entretanto, dentre os trabalhos verificados, ainda há uma ausência de aplicações que estejam correlacionadas com mobilidade urbana, explorando dados reais de uma cidade, e utilizando funções mais avançadas de geoprocessamento (como o uso de geometrias). Do ponto de vista acadêmico, a combinação de problemas urbanos práticos com temas teóricos incentivam a participação e instigam a curiosidade dos alunos.

2.2. *Carpooling*

O *carpooling* é um termo utilizado para o uso compartilhado de um automóvel, para aumentar a eficiência, de um modo geral, de uma rede de transportes[Fu et al. 2008]. Nos Estados Unidos houve um movimento pioneiro no sentido das propostas para *carpooling* e elas já existem há algum tempo [Fu et al. 2008]. Em San Francisco, uma parceira de empresas privadas e setor público desenvolvem o projeto do TRAVINFO [Markowitz 1993], desde 1992. O artigo descreve que a arquitetura usada no sistema inclui 3 bancos de dados integrados: referências geográficas, informações de operações de tráfego e informações de trânsito públicas. Inicialmente ⁷ desenvolvido para tráfego, trânsito, bicicletas e compartilhamento de caronas, foi posteriormente utilizado para inúmeros aplicativos (devido a licença livre dos dados). É possível interagir com o site através de vários tipos de dados, usando APIs (XML), ou Java. Entretanto, a aplicação não disponibiliza a utilização de algoritmos específicos, como caminho mínimo ou caixeiro viajante.

A eficiência do *carpool* também foi comprovada para chamadas de táxi [Liu et al. 2010, Chen et al. 2010]. Porém, alguns problemas podem ser enfrentados na elaboração destes sistemas [Dillenburg et al. 2002]: precisão do GPS e complicações do modelo relacional de SGBDs que não lidam de forma otimizada para constantes mudanças de dados como a localização de objetos em movimento (*moving objects databases* [Trajcevski et al. 2004]). Nota-se nos artigos mais recentes que estes dois temas são discutidos e soluções adicionais são buscadas [Chen et al. 2010, Bandara and Dias 2009].

⁷<http://traffic.511.org> Acesso em 02/06/2014.

Há algumas aplicações brasileiras que já abordam o problema das caronas: o Carona Solidária⁸ trabalha em conjunto com a API do Google Maps para agendar caronas para o Brasil inteiro. Já o software Caronetas⁹ é voltado para empresas, necessitando que a empresa entre em contato com o site para realizar um cadastro e permitir cadastros específicos por meio de e-mails corporativos confirmados. Há ainda comunidades¹⁰ na rede social Facebook¹¹ para agendamento de caronas.

Trabalhos vêm sendo desenvolvidos em outras universidades [Collotta et al. 2012, Junior and Fusco 2013], em especial na UFPR, onde material de divulgação e vagas especiais de parada para caroneiros foram criadas em 2012 mesmo sem ainda terem um sistema de informação para facilitar o agendamento e contato dos usuários. Menciona-se como meta para os próximos anos o desenvolvimento de uma aplicação [Junior and Fusco 2013], e o atual projeto pode abrir portas para uma comunicação e integração do sistema entre as duas universidades.

3. O Protótipo

Para exemplificar a problemática, criamos um protótipo, baseado na arquitetura cliente-servidor (Figura 2). Inicialmente, o usuário (motorista) conecta-se em uma página web e cadastra passageiros, um raio de atuação, e o número de caronas disponíveis. O servidor identifica passageiros dentro de um raio (utilizando funções do PostGIS), e utiliza a API do Google Maps para traçar um caminho entre os pontos mais próximos, gerando uma estimativa da rota.

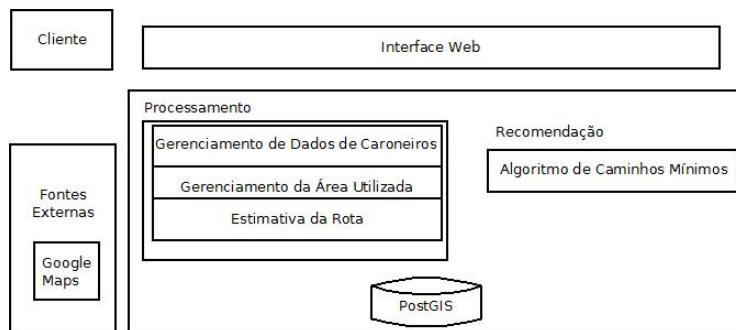


Figura 2. Arquitetura conceitual.

3.1. Esquema do banco de dados

O esquema do banco de dados (apresentado pelo PostGIS na Figura 2) se concentra em uma tabela, como ilustrado na Figura 3-A. A coluna *email* é utilizada como login do usuário, e as colunas *lat* e *long* representam respectivamente a latitude e a longitude, em graus, de um ponto geográfico que identifica a localização da residência de cada usuário. A coluna *endereco* pode ser utilizada para receber um endereço na forma de CEP ou

⁸<http://www.caronasolidaria.com> Acesso em: 12/11/2013

⁹<http://www.caronetas.com.br> Acesso em: 12/11/2013

¹⁰UFPRUTFPR CARONAS <https://www.facebook.com/groups/401201029932308/>
Acesso em: 12/11/2013 e Caronas Centro x UTFPR <https://www.facebook.com/groups/443596625715660/> Acesso em: 12/11/2013

¹¹<http://www.facebook.com.br> Acesso em: 12/11/2013

escrita (tipo Rua Sete de Setembro número 1020 Curitiba, Paraná, Brasil). Este endereço pode ser transformado em latitude e longitude pela *API do Google Maps*. Uma outra possibilidade incluiria (i) considerar pontos fixos (como padarias, postos de gasolina, etc.) para a captação dos caroneiros; e a (ii) disponibilidade de 'check in' e 'check out' de motoristas e caroneiros.

	[PK] integer	nome character varying	email character varying	endereco character varying	lat double precision	long double precision
1	1	Tiago Marti	tiago@gmail	-25.4406449	-49.2465598	
2	3	Referencia	ref3@gmail.	-25.498446	-49.332447	
3	4	Referencia	ref4@gmail.	-25.542132	-49.269619	
4	5	Referencia	ref5@gmail.	-25.422191	-49.271679	
5	6	Referencia	ref6@gmail.	-25.423431	-49.265842	
6	7	Referencia	ref7@gmail.	-25.450715	-49.212971	
7	8	Referencia	ref8@gmail.	-25.41661	-49.281292	
8	9	Referencia	ref9@gmail.	-25.442655	-49.279919	
9	10	Referencia	ref10@gmail	-25.383115	-49.185162	
10	11	Referencia	ref11@gmail	-25.399243	-49.24593	
11	12	Referencia	ref12@gmail	-25.408857	-49.275455	

```

SELECT nome,
       email,
       lat,
       long,
       st_distance_sphere(st_geomfromtext('POINT(' || long || ' ' || lat || ')'),
                           st_centroid(st_geomfromtext('LINESTRING(-49.2465598 -25.4406449, -49.269614 -25.439355)')));
FROM referencia
WHERE
    st_distance_sphere(st_geomfromtext('POINT(' || long || ' ' || lat || ')'),
                       st_centroid(st_geomfromtext('LINESTRING(-49.2465598 -25.4406449, -49.269614 -25.439355)')))< 4000
    AND id NOT IN (1,2)
    ORDER BY $1
LIMIT 8;

```

(A)

(B)

Figura 3. (A) Representação da tabela criada no banco de dados. (B) Consulta dos passageiros mais próximos.

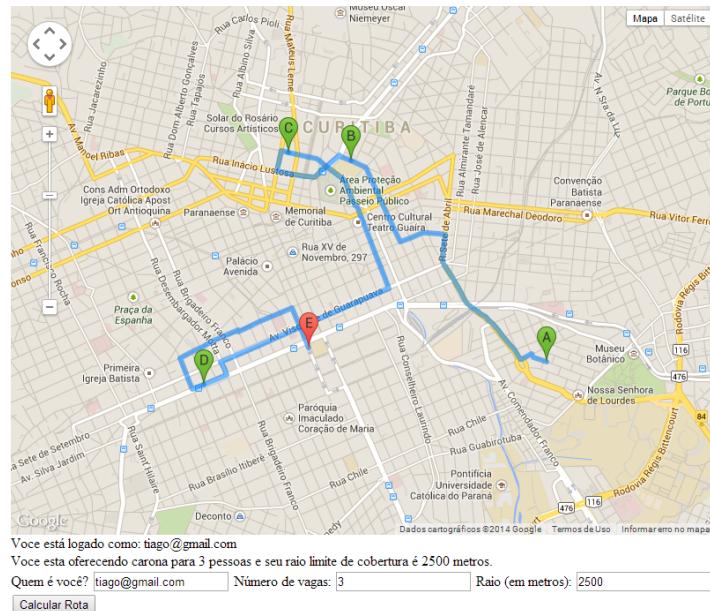


Figura 4. Interface do Protótipo.

3.2. O cálculo dos passageiros mais próximos

O cálculo dos passageiros mais próximos (representado pelo processamento e recomendação na Figura 2) tira proveito de funções em banco de dados geográficos (combinadas com algoritmos) para permitir consultas mais avançadas. Como exemplo, considere o cálculo dos caroneiros mais próximos, verificando quais candidatos a passageiros serão considerados. Esta consulta é realizada pelo PostGIS através do SQL identificado na Figura 3-B (utilizando funções *st_distance_sphere* and *st_centroid* do PostGIS).

O princípio do cálculo é descobrir o centróide entre a residência do motorista e a UTFPR, traçar um círculo com o raio informado pelo usuário e retornar os usuários,

em ordem crescente de maior proximidade com o centróide do ponto de partida do motorista. Esta abordagem foi escolhida pela simplicidade do seu código e facilidade de implementação.

3.3. Interface Web

O cliente, após logar no sistema (representada pela camada de Interface Web na Figura 2), deve preencher 3 campos, como ilustrado na Figura 4. O primeiro campo deverá receber o login do usuário a dar a carona a fim de resgatar seus dados no banco de dados. O segundo campo deverá ser preenchido com o número de vagas ofertadas pelo motorista. O terceiro com o raio de alcance em metros que o motorista deseja para o aplicativo localizar supostos passageiros.

Na Figura 4 tem-se que o ponto A representa as coordenadas do usuário referente ao e-mail inserido. O ponto E representa a UTFPR, o destino final do trajeto. Os pontos B,C,D representam os passageiros mais próximos detectados pela consulta (respeitando o limite de 3 vagas) e o raio de 2500 metros. Ou seja: basicamente atuamos na lógica de quais caroneiros utilizar, tirando proveito de funções geométricas e ranqueamento.

3.4. Outros detalhes e limitações da aplicação e das ferramentas utilizadas

Dentre as vantagens da API do Google Maps, podemos citar: (i) possui um algoritmo de solução do problema do caixeiro-viajante¹²; (ii) possui um algoritmo que calcula o menor caminho entre 2 pontos considerando o tráfego, velocidade máxima das vias e estimativas de tempo da viagem [Brumitt 2007]; e (iii) inclui os seguintes modos de deslocamento: carro, bicicleta, e caminhada. Dentre as desvantagens, podemos citar: (i) o limite de 25000 consultas de mapas por dia que são controlados através de uma chave atrelada a uma conta Google; (ii) o limite de 8 pontos de referência entre a origem ou destino, em outras palavras o software consegue processar até 8 pontos de referência entre o ponto de partida do motorista e o ponto final; (iii) mais consultas ou limites maiores devem requisitar uma licença paga; e (iv) não necessariamente todos os pontos de uma cidade podem estar inclusos no raio de cobertura.

A solução utilizou HTML, PHP, PostgreSQL versão 9.3.2 x64, PostGIS versão 2.1.1 r12113, e Google Maps API v3. Alternativas como Open Street Maps¹³, também podem limitar o número de acessos. Note que no contexto especificado (cidade de Curitiba), não existe nenhuma outra forma de tirar proveito de dados reais de tráfego, horários de pico, rotas alternativas devido a pontes interditadas, etc.

3.5. Dificuldades/Desafios

Dentre as dificuldades e desafios ao tratar de uma aplicação de *Carpool* no contexto de uma comunidade universitária brasileira, podemos citar:

1. Integração de parâmetros diferenciados, como o raio de abrangência do cálculo, número de caroneiros, hora de chegada e saída, capacidade do carro, cálculo de rotas alternativas, etc. O desafio encontra-se em como integrar todos os parâmetros de forma rápida e com um algoritmo eficiente.

¹²<https://developers.google.com/maps/documentation/javascript/directions?\#Waypoints> Acesso em: 18/02/2014

¹³www.openstreetmap.org/#map=11/-25.4946/-49.2867 Acesso em: 18/02/2014

2. Questões de cunho pessoal, como a definição de horários fixos, problemas com cigarro, comida, higiene, pontualidade, segurança, velocidade média, música, senso comum, cortesia, uso de celulares, definição da divisão de despesas, dificuldade de se relacionar com estranhos, etc. Mesmo já existindo uma série de sistemas em funcionamento com foco em *carpool*, nota-se que muitas pessoas não participam, e uma das causas pode ser o receio de se relacionar com estranhos [Selker and Saphir 2010, Chen et al. 2011]. Algumas aplicações¹⁴ sugerem que as pessoas se encontrem antes do processo, para acertar detalhes e discutir opções. Em particular, no contexto de comunidades universitárias, o registro acadêmico do aluno, junto com informações como pontuação das caronas poderiam ser úteis para garantir um nível menor de rejeição.
3. Questões de GIS, como agregações espaço-temporais, reconhecimento de padrões de mobilidade, conexões multimodais (ônibus, navio, carro), heterogeneidade de fontes e dados, conexão com áreas de engarrafamento, altimetria, ruas, pontes, rios, pontos de ônibus, linhas e seus horários. A abordagem da centróide apresentada, por exemplo, traz alguns aspectos negativos: quando o raio escolhido é pequeno, a área de busca é limitada a regiões no entorno do centróide, ignorando outras que também seriam interessantes. Uma abordagem alternativa a centróide envolveria traçar uma trajetória entre o motorista e o destino e criar a área de abrangência com base na união das áreas de um círculo andando com seu centro sobre a reta. Outra alternativa seria criar a área de abrangência de forma semelhante à explicada anteriormente mas sobre as linhas que representam o trajeto original do motorista. Nestas abordagens, funções do PostGIS poderiam facilmente resolver a questão, do ponto de vista teórico. Entretanto, do ponto de vista prático, estas abordagens que envolvem a área de círculos podem ser problemáticas caso haja um rio com pontes distantes, fronteiras de países ou qualquer outro obstáculo geográfico ou político dentro da área considerada. Para evitar estes problemas seria necessário limitar os caroneiros por mais parâmetros, podendo este ser a distância do trajeto à carro entre os dois pontos, excluindo assim os pontos com trajetos muito longos.
4. Questões teóricas [Hartman et al. 2014], como o pareamento de lugar origem e destino, tratamento do problema como grafo bipartido ou problema np-difícil, adicionar ao cálculo variáveis como preferências de passageiros, passageiros VIP, etc. A maioria dos aplicativos citados neste trabalho ainda não permitem a exploração de problemas tradicionais da teoria de grafos (como o cálculo do caminho mínimo), o número de caronas disponíveis, opções de trajeto considerando tráfego intenso, pontes com problemas, trocas de ônibus, raios de abrangência, etc. Dentro do contexto averiguado (cidade de Curitiba), haveria ainda o problema de não haver nenhuma fonte com dados públicos online sobre tráfego, horário de pico, pontos de ônibus, etc.
5. Questões de aspecto dinâmico, como o pareamento de origem e destino em curta janela de tempo, localização espacial rápida de motoristas e caroneiros, algoritmos temporais que permitam conexões multimodais, representação de trajetória em objetos em movimento [Trajcevski et al. 2004], etc.
6. Questões relacionadas ao software, como interface fácil de manusear, processa-

¹⁴<http://rideshare.511.org/carpool/> Acesso em 02/06/2014.

mento rápido e eficiente (podendo ser adaptável a dispositivos diferenciados), possibilidade de adicionar sugestões de melhora e conexões com redes sociais.

Em particular, para a cidade de Curitiba, também podemos citar como desafio a inclusão na parametrização de aspectos intrínsecos da cidade, como o uso de biarticulados, exploração dos eixos estruturais, porcentual de dias chuvosos (onde o uso de veículos é maior); etc. Além disso, apesar da cidade disponibilizar recentemente o acesso a seus dados, ainda há um longo processo a ser explorado na direção de aplicativos e sua posterior utilização pela comunidade.

4. Conclusão e Trabalhos Futuros

Este trabalho apresentou os conceitos, as aplicações, um protótipo e os desafios para *carpooling*, utilizando o contexto de uma comunidade universitária brasileira. Através da abordagem de uma arquitetura cliente-servidor, vários parâmetros são ilustrados, como a falta de dados, a dificuldade de aplicar algoritmos com aplicações diversas, além de alguns problemas de fundo social (como a desconfiança de compartilhar caronas com desconhecidos). Dentre os trabalhos futuros podemos citar a exploração de outros banco de dados (como os baseados em grafos), o aprimoramento da interface para dispositivos móveis, a navegação com domínios diferentes de dados, caminhos entre ambientes heterogêneos, além de testes em ambientes diversificados.

Agradecimentos Os autores agradecem o CNPq, CAPES, Secretaria de Estado da Ciência, Tecnologia e Ensino Superior do Paraná e Fundação Araucária pelo apoio financeiro.

Referências

- Bandara, H. A. N. C. and Dias, D. (2009). A multi-agent system for dynamic ride sharing. In *ICIIS' 09*, pages 199–203.
- Braga, J. O. N., Costa, L. A. d., Guimarães, A. L., and Tello, J. C. R. (2008). O uso do geoprocessamento no diagnóstico dos roteiros de coleta de lixo da cidade de Manaus. *Engenharia Sanitária e Ambiental*, 13:387 – 394.
- Brumitt, B. (2007). The road to better path-finding, <http://googleblog.blogspot.com.br/2007/11/road-to-better-path-finding.html>. Google Official Blog.
- Chen, C.-M., Shallcross, D., Shih, Y.-C., Wu, Y.-C., Kuo, S.-P., Hsi, Y.-Y., Holderby, Y., and Chou, W. (2011). Smart ride share with flexible route matching. In *13th International Conference on Advanced Communication Technology (ICACT)*, pages 1506–1510.
- Chen, P.-Y., Liu, J.-W., and Chen, W.-T. (2010). A Fuel-Saving and Pollution-Reducing Dynamic Taxi-Sharing Protocol in VANETs. In *VTC 2010-Fall, 2010 IEEE 72nd*, pages 1–5.
- Collotta, M., Pau, G., Salerno, V., and Scata, G. (2012). A novel trust based algorithm for carpooling transportation systems. In *Energy Conference and Exhibition (ENERGY-CON), 2012 IEEE International*, pages 1077–1082.

- Dillenburg, J., Wolfson, O., and Nelson, P. (2002). The intelligent travel assistant. In *IEEE ITSC' 02*, pages 691–696.
- Fu, Y., Fang, Y., Jiang, C., and Cheng, J. (2008). Dynamic ride sharing community service on traffic information grid. In *ICICTA '08*, volume 2, pages 348–352.
- Graziotin, D. (2013). An analysis of issues against the adoption of dynamic carpooling, arxiv preprint arxiv:1306.0361.
- Hamada, E. and do Valle Gonçalves, R. R. (2007). *Introdução ao Geoprocessamento: princípios básicos e aplicação*. EMBRAPA.
- Hartman, I. B.-A., Keren, D., Dbai, A. A., Cohen, E., Knapen, L., Yasar, A.-U.-H., and Janssens, D. (2014). Theory and practice in large carpooling problems. *Procedia Computer Science*, 32(0):339 – 347.
- Junior, R. M. and Fusco, R. (2013). Projeto Carona Solidária na UFPR. *Revista Latino-Americana de Inovação e Engenharia de Produção*, 1:136–143.
- Liu, N., Liu, M., Cao, J., Chen, G., and Lou, W. (2010). When transportation meets communication: V2p over vanets. In *IEEE ICDCS' 10*, pages 567–576.
- Markowitz, J. (1993). TravInfo. The San Francisco Bay Area intermodal traveler information system. In *WESCON '93*, pages 339–344.
- Megalingam, R. K., Nair, R., and Radhakrishnan, V. (2011). Automated wireless carpooling system for an eco-friendly travel. In *ICECT '11*, volume 4, pages 325–329.
- Miller, H., Wu, Y.-H., and Hung, M.-C. (1999). Gis-based dynamic traffic congestion modeling to support time-critical logistics. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, volume Track6, pages 9 pp.–.
- Rocha, C. H. B. (2002). *Geoprocessamento: Tecnologia Transdisciplinar Equipamentos, Processos e Metodologias*. UFRJ. ISBN 9788590148319.
- Sassi, A., Mamei, M., and Zambonelli, F. (2014). Towards a general infrastructure for location-based smart mobility services. In *High Performance Computing Simulation (HPCS), 2014 International Conference on*, pages 849–856.
- Selker, T. and Saphir, P. (2010). Travelrole: A carpooling / physical social network creator. In *CTS '10*, pages 629–634.
- Sghaier, M., Zgaya, H., Hammadi, S., and Tahon, C. (2011). A novel approach based on a distributed dynamic graph modeling set up over a subdivision process to deal with distributed optimized real time carpooling requests. In *IEEE ITSC '11*, pages 1311–1316.
- Trajcevski, G., Wolfson, O., Hinrichs, K., and Chamberlain, S. (2004). Managing uncertainty in moving objects databases. *ACM Trans. Database Syst.*, 29(3):463–507.
- Yan, S., Chen, C.-Y., and Lin, Y.-F. (2011). A model with a heuristic algorithm for solving the long-term many-to-many car pooling problem. *IEEE ITSC' 11*, 12(4):1362–1373.

Um Estudo sobre Modelagem Lógica para Bancos de Dados NoSQL

Claudio de Lima, Ronaldo S. Mello

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina
Caixa Postal 476 – 88.040-900 – Florianópolis –SC – Brasil

`claudio.lima@posgrad.ufsc.br, r.mello@ufsc.br`

Abstract. *This paper presents an overview of logical modeling approaches for NoSQL data and other complex data models. We analyze conceptual and logical models proposed by related work, and also transformation process between them. At the end of the paper, we suggest some guidelines for a methodology for a document NoSQL database logical design that considers the contributions of these works.*

Resumo. *Este artigo apresenta uma visão geral das abordagens para modelagem lógica para dados NoSQL e para outros modelos de dados complexos. Os modelos conceituais e lógicos propostos por trabalhos relacionados são analisados, assim como os processos de transformação entre os modelos. Ao final do trabalho, sugere-se algumas diretrizes para uma metodologia para o projeto lógico de banco de dados NoSQL de documentos, considerando as contribuições destes trabalhos.*

1. Introdução

Computação em nuvem é um paradigma que visa prover serviços sob demanda e com pagamento baseado no uso, considerando desde o usuário final que hospeda seus documentos pessoais na Internet até empresas que terceirizam toda infraestrutura de tecnologia da informação. Neste contexto, a imensa quantidade de dados gerados diariamente em vários domínios de aplicação, como por exemplo, na Web e em redes sociais, traz grandes desafios na forma de manipulação, armazenamento e processamento de consultas em várias áreas da computação, e em especial na área de Banco de Dados (BD) [Vieira et al. 2012].

Os Sistemas Gerenciadores de BD (SGBDs) tradicionais não são os mais adequados às necessidades de gerenciamento de dados na nuvem. Uma tendência para solucionar os desafios inerentes a esta problemática é o movimento denominado *NoSQL*, que consiste em SGBDs não-relacionais projetados para gerenciar grandes volumes de dados e que disponibilizam estruturas e interfaces de acesso simples [Sousa et al. 2010]. Os BDs NoSQL proporcionam um grande número de operações de leitura e escrita por segundo, característica comum em aplicações Web modernas [Cattell 2010]. O suporte a tipos de dados complexos, semiestruturados ou não estruturados também favorece o uso destes BDs, que atualmente são categorizados em *chave-valor*, *documento*, *família de colunas* e *baseado em grafos* [McMurtry et al. 2013].

A organização dos dados em BDs NoSQL requer significativas decisões de projeto, uma vez que elas afetam os requisitos principais de qualidade, incluindo escalabilidade, desempenho e consistência [Bugiotti 2014]. O projeto tradicional de BDs é um processo constituído por três fases de modelagem de dados ([Elmasri 2011], [Batini 1992]): *conceitual*, *lógica* e *física*. Inicialmente, na etapa de modelagem conceitual um esquema representando as informações de um domínio é representado em um modelo de alto nível de abstração. Na sequência, na modelagem lógica, o esquema conceitual é transformado em um esquema de mais baixa abstração adequado ao modelo de dados alvo, ou seja, o modelo no qual o BD será fisicamente implementado.

Metodologias e ferramentas de suporte ao projeto de BDs NoSQL é um tópico ainda pouco explorado na literatura de BDs [Atzeni 2012]. No contexto acadêmico, poucos trabalhos propõem metodologias de projeto lógico para BDs NoSQL, com ênfase em processos que convertam modelagens conceituais para representações lógicas em modelos de dados NoSQL ([Bugiotti 2014], [Jovanovic 2013]). Este artigo realiza uma análise comparativa destas abordagens com o intuito de apresentar um estado da arte sobre o tema. Além disso, abordagens que produzem esquemas não-relacionais no nível lógico, como esquemas XML ([Schroeder 2008], [Choi 2003], [Fong 2006], [Elmasri 2002], [Bird 2000]) e esquemas Orientados a Objetos (OO) ([Nachouki 1991], [Biskup 1995], [Elmasri 1993], [Narasimhan 1993], [Fong 1995]) também são analisados com o objetivo de identificar contribuições que possam ser aplicadas no embasamento de uma proposta para o projeto lógico de BDs NoSQL de documento.

Este artigo é organizado em mais três seções. Uma visão geral e as principais contribuições das abordagens para modelagem lógica de dados recém-mencionadas são apresentadas na seção dois. Na seção três, em linhas gerais, algumas diretrizes são sugeridas para uma metodologia para o projeto lógico de BDs NoSQL documento. Finalmente, a seção quatro apresenta as considerações finais sobre o tema em questão.

2. Modelagem de Dados para NoSQL

Os BDs NoSQL estão disponíveis em uma variedade de formas e funcionalidades, sendo atualmente categorizados, conforme o modelo de dados utilizado, em *chave-valor*, *documento*, *família de colunas* e *baseado em grafos* [McMurtry et al. 2013]. O modelo *chave-valor* é o mais simples e consiste essencialmente em uma grande *tabela hash* onde um valor é associado a uma chave única que é utilizada para recuperar os dados no BD. *Voldemort* e *Riak* são exemplos de BDs desta categoria.

O modelo de dados *família de colunas* organiza seus dados em linhas e colunas e a sua grande característica encontra-se na sua abordagem desnormalizada para estruturar dados esparsos. BDs de família de colunas podem ser vistos como a exploração de dados tabulares com a divisão das colunas em grupos conhecidos como *família-coluna*. *Cassandra* e *Hbase* são exemplos de BDs de *família de colunas*. Os BDs NoSQL *baseados em grafos* têm como foco principal os relacionamentos que entidades têm umas com as outras. Eles armazenam *nós*, que são instâncias de entidades, e *arestas*, que especificam os relacionamentos entre os nós. A principal finalidade desta categoria de BDs é permitir que uma determinada aplicação execute eficientemente consultas que atravessam uma rede de nós e arestas, e analise os relacionamentos entre as entidades. *Neo4J* e *Infinite Graph* são exemplos de BDs NoSQL baseados em grafos.

A categoria de BDs NoSQL *documento* é semelhante em conceito a um BD chave-valor, exceto pelo fato que os valores armazenados são documentos. Um documento é um conjunto de campos e valores nomeados (pares chave-valor), sendo que cada valor de campo pode conter um *item escalar simples* ou um *item composto*, tal como uma *lista* e um *documento aninhado*. Os dados nos campos em um documento podem ser codificados em uma variedade de maneiras, incluindo JSON, BSON, XML, YAML, ou mesmo armazenados como texto simples. De acordo com [Kaur 2013], os BDs orientados a documento formam uma categoria apropriada para aplicações Web que envolve o armazenamento de dados semiestruturados e a execução de consultas dinâmicas. Estes BDs são capazes de suportar escalabilidade horizontal, proporcionam alta disponibilidade e são flexíveis quanto ao suporte a dados. *MongoDB*, *RavenDB* e *CouchDB* são exemplos de BDs NoSQL de documento.

Os BDs NoSQL não requerem um esquema associado aos dados, e por esse motivo são considerados adequados a aplicações do tipo *data-driven*, ou seja, concentram-se no modelo de consulta e constroem o modelo de dados ao redor dele, a fim de satisfazerem suas necessidades de forma eficiente. Entretanto, estes dados apresentam alguma estrutura, e obter vantagens desta estrutura torna-se muitas vezes necessário. A persistência de dados de aplicações deve ser mapeada para itens de dados modelados (elementos), como documentos e pares chave-valor, disponíveis no sistema destino, e, portanto, a organização dos dados em BDs NoSQL requer significativas decisões de projeto. Estas decisões afetam os requisitos principais de qualidade, incluindo escalabilidade e desempenho, bem como a consistência [Bugiotti 2014].

O projeto tradicional de BDs é um processo com três fases de modelagem de dados, os projetos *conceitual*, *lógico* e *físico*. Enquanto o objetivo do projeto conceitual é produzir um esquema expressivo e capaz de representar os dados de um domínio de informação, o objetivo do projeto lógico é transformar um esquema conceitual em uma representação equivalente em um modelo lógico que se aproxima do modelo de implementação do BD. No meio acadêmico, poucos trabalhos propõem metodologias de projeto lógico para BDs NoSQL, com ênfase em processos que convertam modelagens conceituais para representações lógicas em modelos de dados NoSQL. Neste contexto, abordagens relevantes para a modelagem de dados NoSQL e que utilizam esquemas não-relacionais, como esquemas XML e OO, são revistas nesta seção visando identificar contribuições para a modelagem lógica de BDs NoSQL. Modelos XML e OO, assim como os modelos de dados NoSQL, são modelos de dados (ou objetos) complexos, cujas similaridades de estratégias de mapeamento, como o tratamento de atributos multivvalorados e aninhados, podem ser adaptadas para o projeto lógico de BDs NoSQL.

Para fins de análise e classificação, os trabalhos relacionados são separados em três grupos de acordo com o modelo lógico: (i) Modelo lógico NoSQL; (ii) Modelo lógico XML; e (iii) Modelo lógico OO. A partir desta revisão, uma análise comparativa entre os trabalhos é realizada com o objetivo de relacionar os níveis de modelagem (conceitual, lógico e físico) atendidos por cada proposta.

2.1. Modelo Lógico NoSQL

O trabalho de [Bugiotti 2014] apresenta uma abordagem lógica para o problema de projeto para BD NoSQL, chamada de *NoAM*, que explora os pontos em comum de

algumas categorias de BDs NoSQL. A proposta é baseada em um modelo de dados abstrato intermediário, em nível lógico, utilizado para representar os dados de aplicações como coleções de objetos agregados, e demonstra como a representação intermediária pode ser implementada em alguns BDs NoSQL, levando em conta as suas características específicas. O modelo lógico NoSQL proposto é baseado no conceito de *agregados*, termo da área de *Domain-Driven Design* (DDD) [Evans 2003]. DDD é uma abordagem de projeto OO amplamente adotada, sendo um agregado uma coleção de objetos relacionados, de forma aninhada, que pode ser tratado como uma unidade [Sadalage 2013]. A abordagem sugere um suporte eficaz para escalabilidade, consistência e desempenho e baseia-se em quatro fases principais.

A primeira fase trata do *projeto de agregados*, que visa identificar as várias classes de objetos agregados necessários em uma aplicação. Essa atividade é conduzida por casos de uso e requisitos funcionais. A fase seguinte trata do *particionamento de agregados*, sendo que os agregados são divididos em elementos de dados menores. Essa atividade é conduzida por casos de uso e requisitos de desempenho. Na sequência, na fase de *projeto de BD NoSQL de alto nível*, os agregados são mapeados para o modelo de dados intermediário *NoAM*, de acordo com as partições identificadas. A última fase é a *implementação*, que converte a representação intermediária de dados para os elementos de modelagem específicos do BD destino.

O trabalho de [Jovanovic 2013] utiliza IDEF1X (*Integration DEFinition for Information Modeling*) na modelagem conceitual para representar o domínio de uma aplicação, e também para representar o modelo lógico para NoSQL baseado em agregados, obtido através de um processo de conversão entre os modelos. Esta proposta fornece suporte para análise de diferentes formas de modelagem, como por exemplo, o particionamento do esquema de dados em agregados menores e independentes para o contexto do SOA (*Service Oriented Architecture*). Em SOA, aplicações são organizadas em pequenos serviços, reduzindo a modelagem de dados para peças pequenas e independentes em torno de um objetivo. O modelo lógico proposto pode ser utilizado em BDs NoSQL nas categorias chave-valor, documento e família de colunas.

2.2. Modelo Lógico XML

A literatura apresenta muitos trabalhos relacionados a metodologias de projeto para BDs XML. O trabalho de [Schroeder 2008] apresenta uma abordagem de projeto de geração de esquemas XML a partir de esquemas conceituais representados no modelo Entidade-Relacionamento Estendido (EER), considerando a carga de trabalho esperada da aplicação. O esquema conceitual é traduzido em um esquema lógico XML na etapa de projeto lógico. O processo de conversão aplica um conjunto de regras para o mapeamento de cada construtor conceitual em uma representação equivalente no modelo lógico XML. Este modelo lógico é um modelo abstrato para representar os diferentes modelos de implementação XML (DTD e XML Schema). A consideração da carga de trabalho visa para produzir esquemas XML que minimizam o impacto das relações de referência sobre o desempenho das principais operações de acesso.

Metodologias para produzir esquemas XML em DTD a partir de esquemas EER são apresentadas em Fong et. al. [Fong 2006] e Choi et. al. [Choi 2003]. A metodologia proposta por [Fong 2006] considera grande parte dos construtores conceituais do EER e

susas restrições para gerar um esquema XML em DTD. Ela endereça o problema de aplicações que desejam manter seus dados em BDs relacionais e precisam trabalhar com estes dados no formato XML no nível da aplicação. Por este motivo, uma abordagem de engenharia reversa é utilizada para transformar um esquema relacional em um esquema EER que posteriormente é convertido em uma DTD. O trabalho de [Choi 2003] propõe uma metodologia de projeto unificado para XML, em que um esquema EER é transformado diretamente em uma especificação em DTD, que por sua vez é transformada em um esquema físico.

O trabalho de Elmasri et. al. [Elmasri 2002] apresenta uma metodologia para a conversão de esquemas EER em estruturas hierárquicas. Um algoritmo é proposto para a geração de visões hierárquicas customizadas a partir de uma modelagem EER. Inicialmente, eventuais ciclos de um esquema EER são removidos e, na sequência, o usuário informa uma entidade de partida para a geração de uma estrutura hierárquica. Já a abordagem de [Bird 2000] visa reduzir a redundância de dados e aumentar a conectividade das instâncias XML resultantes. Ela propõe um projeto de BD em três níveis, sendo que no nível conceitual são utilizados elementos da UML. No nível lógico, diagramas de classe são estendidos com estereótipos, especificando conceitos específicos do modelo de dados XML. A modelagem explícita de conceitos XML através de diagramas UML estendidos permite definir regras de conversão para o nível de implementação, utilizando o *XML Schema*.

2.3. Modelo Lógico OO

Muitos trabalhos apresentam processos de conversão de modelagens conceituais para representações lógicas OO. Na abordagem de [Nachouki 1991], o processo inicia com a criação de um esquema OO inicial a partir de um diagrama ER. Na sequência, *caminhos de acesso* lógico de operações no esquema inicial são determinados, e para cada caminho constrói-se uma árvore de consulta. Os caminhos de acesso são então fundidos e resultam em um esquema acíclico. Finalmente, as classes do esquema são mescladas com os caminhos, e as classes resultantes podem ser codificadas na linguagem do SGBDOO O₂. O trabalho de [Biskup 1995] utiliza *F-Logic*, um formalismo baseado em lógica para especificar o raciocínio sobre conceitos OO, complementado por uma noção de restrições semânticas para a sua transformação. Um esquema ER é primeiro transformado num esquema OO *abstrato* em F-Logic, o qual é refinado através da decomposição e mescla de classes. O esquema resultante é então mapeado para um esquema OO *concreto*, utilizando o SGBDOO ONTOS.

A abordagem de [Elmasri 1993] utiliza o modelo EER como partida e um modelo OO *virtual* como alvo, enfatizando a transformação de diferentes tipos de construções de generalização e especialização. Além do mapeamento de estruturas EER, o trabalho aborda a geração automática e a integração de métodos que impõem restrições de integridade no esquema EER. O trabalho de [Narasimhan 1993] enfatiza a integração das restrições definidas por um esquema ER em um esquema OO, além da transformação estrutural. Ela sugere a criação de uma classe especial de restrição com uma subclasse para cada classe no esquema OO, obtendo-se, assim, duas hierarquias de classes, uma para a estrutura, e a outra para os métodos de restrição.

O trabalho de [Fong 1995] apresenta uma metodologia de reengenharia de um modelo EER existente para o modelo OMT (*Object Modeling Technique*), utilizando um conjunto de regras de tradução do modelo EER para um modelo genérico OO em OMT. Argumenta-se que, do ponto de vista do usuário, o modelo EER é mais compreensível que o modelo OMT devido a sua simplicidade. Desta forma, adota-se o método tradicional de projeto de BDs iniciando pela modelagem EER e sua posterior conversão para um esquema OMT como parte do projeto de um BDOO.

2.4. Comparativo

O quadro comparativo mostrado na Tabela 1 situa cada uma das abordagens apresentadas nas etapas de projeto conceitual, lógico e de implementação de um BD, descrevendo os respectivos modelos utilizados em cada uma delas.

Tabela 1. Comparativo dos Trabalhos Relacionados.

Abordagem	Projeto de Banco de Dados		
	Conceitual	Lógico	Implementação
Modelo Lógico NoSQL			
[Bugiotti 2014]	UML	NoAM baseado em agregados	Elementos específicos de modelos de BDs NoSQL
[Jovanovic 2013]	IDEF1X	IDEF1X baseado em agregados	-
Modelo Lógico XML			
[Schroeder 2008]	EER	XML Lógico	DTD / XML Schema
[Choi 2003]	EER	-	DTD
[Fong 2006]	EER	-	DTD
[Elmasri 2002]	EER	Estruturas hierárquicas	XML Schema
[Bird 2000]	UML	UML+ estereótipos	XML Schema
Modelo Lógico OO			
[Nachouki 1991]	ER	Esquema OO Lógico	DDL O ₂
[Biskup 1995]	ER	F-Logic	DDL ONTOS
[Elmasri 1993]	EER	Esquema OO	-
[Narasimhan 1993]	ER	Esquema OO	-
[Fong 1995]	EER	OMT	-

Quanto aos trabalhos relacionados a metodologias de projeto para BDs NoSQL, observa-se que eles propõem esquemas lógicos utilizando o conceito de agregado no nível lógico. De acordo com o trabalho de [Sadalage 2013], agregados formam os limites para as operações ACID com os BDs, e os modelos de dados NoSQL das categorias *chave-valor*, *documento*, e *família de colunas* são considerados BDs orientados a agregados. O trabalho de [Bugiotti 2014] explora os pontos em comum de algumas categorias de BDs NoSQL e, embora atue nas três etapas de projeto, não explora a utilização completa de construtores conceituais para a modelagem de um domínio de aplicação nem formaliza processos de conversão entre modelagens conceituais e representações lógicas no modelo NoAM. A proposta de [Jovanovic 2013] apresenta a conversão de um modelo conceitual em uma modelo lógico, ambos utilizando a linguagem IDEF1X, e não aborda a etapa de modelagem física.

Em relação aos trabalhos relacionados de metodologias de projeto para BDs XML, observa-se que a proposta de [Schroeder 2008] atua nas três etapas de projeto e considera todos os construtores do modelo conceitual EER para a geração de uma

estrutura XML equivalente. Informações referentes à carga estimada para o BDs são utilizadas para efetuar otimizações na estrutura XML durante o processo de transformação. No trabalho de [Elmasri 2002], os esquemas hierárquicos e respectivos esquemas XML gerados limitam-se a representar visões sobre um esquema EER, não considerando todas as relações semânticas concebidas na modelagem conceitual. Além disso, o modelo hierárquico utilizado é bastante simplificado, definindo apenas uma estrutura de grafo sem considerar restrições específicas do modelo XML. Já o trabalho de [Bird 2000] propõe uma metodologia de projeto baseada na UML, identificando quais seriam os principais passos de um algoritmo de mapeamento do nível conceitual para o lógico, mas sem propô-lo concretamente. Quanto aos modelos conceituais utilizados, observa-se o grande emprego do ER e do EER.

Por fim, com relação às abordagens que apresentam processos de conversão de modelagens conceituais para representações OO, observa-se que há muitas semelhanças na transformação de construtores do ER para OO. Pequenas diferenças existem na maneira pela qual os tipos de relacionamento binários são representados, e também a respeito de estruturas de generalização consideradas. O trabalho de [Nachouki 1991] considera o processamento de requisitos durante a transformação. A qualidade de um esquema OO é discutida nos trabalhos de [Nachouki 1991] e [Biskup 1995]. O primeiro concentra-se na identificação de classes, enquanto o segundo investiga como um esquema OO resultante de uma transformação inicial pode ser melhorado através da fusão e decomposição de classes. Uma diferença entre as abordagens OO está no tratamento de restrições de integridade, o qual geralmente é superficial. Observa-se também que as abordagens de [Elmasri 1993], [Narasimhan 1993], [Biskup 1995] e [Fong 1995] apresentam metodologias independentes de modelos OO específicos.

3. Diretrizes para Projeto Lógico de BDs NoSQL

Esta seção apresenta, em linhas gerais, diretrizes para uma metodologia para projeto lógico de BDs NoSQL, considerando o modelo de dados de documento e as características das abordagens apresentadas na seção anterior. Inicialmente, ressalta-se que embora não seja requerido que um BD NoSQL possua um esquema padrão, a importância de um modelo de dados associado a eles está no melhor entendimento e na demonstração de como os dados são persistidos no BD alvo [Kaur 2013].

3.1. Modelagem Conceitual

Considerando a modelagem conceitual, o modelo EER possui uma ampla utilização pelos trabalhos relacionados uma vez que é um modelo simples e de caráter mais geral em relação a outros modelos conceituais. Desta forma, sugere-se que o esquema conceitual de entrada para um processo de projeto lógico para BDs NoSQL seja o modelo EER. Dado o seu alto nível de abstração para modelagem de dados, ele pode ser facilmente adaptado para trabalhar com outros modelos conceituais como a UML e o IDEF1X, utilizados pelas abordagens relacionadas à BDs NoSQL.

3.2. Modelagem Lógica

Considerando a modelagem lógica, percebe-se que os trabalhos que propõem esquemas lógicos para NoSQL utilizam o conceito de agregados. A escolha por uma representação lógica baseada em agregados é justificada pelo fato de que eles apoiam os requisitos

típicos dos BDs NoSQL, ou seja, são considerados unidades de distribuição, fornecendo suporte à *escalabilidade* (agregados são distribuídos entre os nós de um cluster, onde cada agregado localiza-se em um único nó), e *consistência*, na medida em que for necessário (em atualizações que abrangem vários agregados, por exemplo, pode-se utilizar consistência eventual), e podem ser divididos em pequenos elementos de dados, permitindo o acesso à porções específicas dos agregados, por questões de *desempenho* [Bugiotti 2014]. Recomenda-se que um BD NoSQL da categoria documento seja projetado considerando o conceito de agregado, pois ele manipula documentos, que concretamente são estruturas de dados *hierárquicas* que podem ter coleções de dados aninhados, além de valores escalares ([Sadlage 2013], [McMurtry 2013]).

3.3. Processo de Conversão

Durante o projeto lógico, entende-se que um esquema conceitual deva ser transformado em um esquema lógico NoSQL através de um processo baseado em regras de conversão com a finalidade de prover estratégias de mapeamento para os construtores do modelo conceitual escolhido. Estas regras devem tratar a conversão de um grafo (como um diagrama EER) em árvores (de agregados), sendo que o modelo lógico NoSQL documento deve ser composto por estruturas de dados na forma de árvores hierárquicas, ideia análoga a abordagens relacionadas na seção anterior.

A Figura 1 apresenta exemplos de esquemas conceitual e lógico envolvidos em um possível processo de conversão. O processo inicia com um esquema EER, no nível conceitual, que é transformado, através de regras de conversão, em um esquema lógico NoSQL documento baseado em agregados. Posteriormente, o esquema lógico é traduzido para o modelo de implementação de um BDs NoSQL documento.

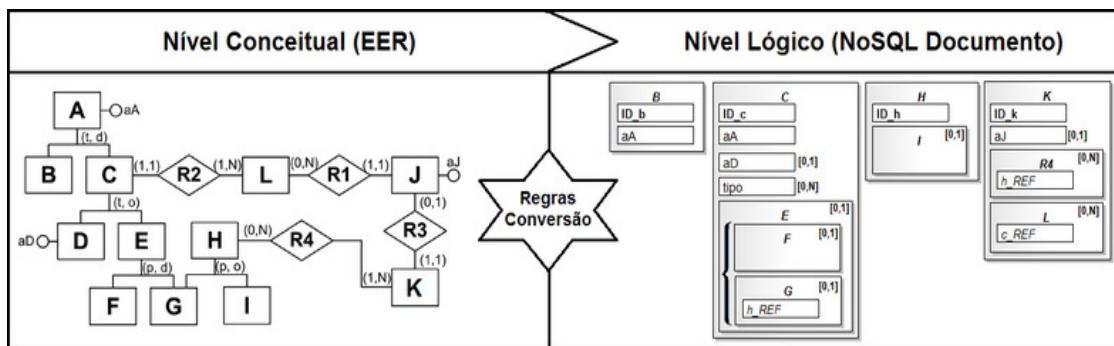


Figura 1. Exemplos de esquemas envolvidos em um processo de conversão.

3.4. Informações de Carga

O modelo lógico para dados NoSQL documento, que utiliza a noção de agregado, deve ser impulsionado por necessidades de consistência e principalmente baseado nas operações de acesso a dados. Desta forma, com a finalidade de otimizar as estruturas lógicas, acredita-se que informações de carga devam ser utilizadas para a geração de uma estrutura NoSQL documento que possa responder de forma eficiente as principais e mais custosas operações do BD NoSQL. Estas otimizações na estrutura devem ser obtidas através de decisões tomadas no mapeamento de estruturas conceituais para estruturas lógicas NoSQL documento, no sentido de não permitir redundância de dados e ao mesmo tempo apresentar uma estrutura adequada para o aninhamento de conceitos frequentemente acessados pela aplicação.

4. Considerações Finais

As necessidades atuais de gerenciamento e processamento de grandes volumes de dados na nuvem tornam evidente a importância da utilização de BDs NoSQL. Estes BDs proporcionam grande número de operações por segundo, característica comum em aplicações Web modernas, e o suporte a tipos de dados complexos, semiestruturados ou não estruturados. A persistência de dados de aplicações deve ser mapeada para elementos modelados, como coleções, documentos e pares chave-valor, disponíveis no sistema destino, e, portanto, a organização dos dados em BDs NoSQL requer significativas decisões de projeto, uma vez que elas afetam os requisitos principais de qualidade, incluindo escalabilidade, desempenho e consistência.

Metodologias consolidadas para a modelagem de BDs assumem que o modelo conceitual deve ser independente de quaisquer modelos de implementação e que a independência entre os modelos do projeto de um BD é um requisito fundamental para garantir a portabilidade da metodologia ([Elmasri 2011, Batini 1992]). Quanto à etapa de modelagem conceitual, uma boa escolha é o modelo EER, por ser um modelo genérico, simples e muito empregado no âmbito acadêmico e comercial para modelagem conceitual de BDs, se comparado com a UML, mais voltada para o projeto de aplicações OO.

Na etapa de modelagem lógica, as abordagens relacionadas destacam a tendência de se utilizar o conceito de agregados, uma vez que estes gravam e recuperam dados organizados de forma hierárquica, que podem consistir em coleções de dados aninhados. O processo de conversão entre os modelos conceitual e lógico trata essencialmente da conversão de um grafo em árvores, ideia análoga das abordagens relacionadas (XML e OO), e que podem contribuir para a construção do processo de conversão para dados de BDs NoSQL documento.

Essa problemática de modelagem de dados NoSQL para a categoria documento e a definição de uma metodologia robusta para projeto lógico de BDs NoSQL documento são o foco de uma pesquisa em andamento no Programa de Pós-Graduação em Ciência da Computação da UFSC. Os estudos apresentados neste artigo estão servindo de base para a construção desta metodologia.

Referências

- Atzeni, P., Bugiotti, F. and Rossi, L. (2012) “Uniform Access to Non-Relational Database Systems: The SOS platform”. In: CAiSE, 2012. p.160–174.
- Batini, C., Ceri, S. and Navathe, S. B. (1992) “Conceptual Database Design: An Entity-Relationship Approach”. Benjamin/Cummings, 1992.
- Bird, L., Goodchild, A. and Halpin, T. (2000) “Object Role Modeling and XML-Schema”. In: ER, 2000. p.661–705.
- Biskup, J., Menzel, R. and Polle, T. (1995) “Transforming an Entity-Relationship Schema into Object-Oriented Database Schemas”. In: ADBIS, 1995. p.109–136.
- Bugiotti, F., Cabibbo, L., Atzeni, P. and Torlone, R. (2014) “Database Design for NoSQL Systems”. In: ER 2014. p.223-231.

- Cattell, R. (2010) “Scalable SQL and NoSQL Data Stores”. SIGMOD Record, 39(4):12–27, 2010.
- Choi, M., Lim, J. and Joo, K. (2003) “Developing a Unified Design Methodology based on Extended Entity-Relationship Model for XML”. In: ICCS, 2003. p.920–929.
- Elmasri, R., James, S. and Kouramajian, V. (1993) “Automatic Class and Method Generation for Object-Oriented Databases”. In: DOOD, 1993, Springer LNCS 760, p.395-414.
- Elmasri, R., Wu, Y., Hojabri, B., Li, C. and Fu, J. (2002) “Conceptual Modeling for Customized XML Schemas”. In: ER, 2002. p.429–443.
- Elmasri, R., Navathe, S. B. (2011) “Fundamentals of Database Systems”. 6th edition, Pearson Addison Wesley, 2011.
- Evans, E. (2003) “Domain-Driven Design: Tackling Complexity in the Heart of Software”. Addison-Wesley, 2003.
- Fong, J. (1995) “Mapping Extended Entity-Relationship Model to Object Modeling Technique”. SIGMOD Record, v. 24, n. 3, p.18-22, 1995.
- Fong, J., Fong, A., Wong, H. K. and Yu, P. (2006) “Translating Relational Schema with Constraints into XML Schema”. International Journal of Software Engineering and Knowledge Engineering, 2006. n.16, p.201–244.
- Jovanovic, V., Benson, S. (2013) “Aggregate Data Modeling”. In: SAIS, 2013.
- Kaur, K. and Rani, R. (2013) “Modeling and Querying Data in NoSQL Databases”. IEEE. In: International Conference on Big Data, 2013. p.1-7.
- McMurtry, D., Oakley, A., Sharp, J., Subramanian, M., Zhang, H. (2013) “Data Access for Highly-Scalable Solutions: Using SQL, NoSQL, and Polyglot Persistence”. Microsoft, 2013. <http://msdn.microsoft.com/en-us/library/dn271399.aspx>.
- Nachouki, J., Chastang, M.P. and Briand, H. (1991) “From Entity-Relationship Diagram to an Object-Oriented Database”. In: ER, 1991. p.459-482.
- Narasimhan, B., Navathe, S. and Jayaraman, S. (1993) “On Mapping ER and Relational Models onto OO Schemas”. In: ER, 1993. p.402–413.
- Sadalage, P. J. and Fowler, M. J. (2013) “NoSQL Distilled”. Addison-Wesley, 2013.
- Sousa, F. R. C., Moreira, L. O., Macêdo, J. A. and Machado, J. C. (2010) “Gerenciamento de Dados em Nuvem: Conceitos, Sistemas e Desafios”. In: SBBD 2010. p. 101–130.
- Schroeder, R. and Mello, R. S. (2008) “Improving Query Performance on XML Documents: A Workload-Driven Design Approach”. In: DOCENG, 2008. p.177-186.
- Vieira, M. R., Figueiredo, J. M., Liberatti, G. and Viebrantz, A. F. M. (2012) “Bancos de Dados NoSQL: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data”. Minicurso SBBD 2012, São Paulo, 2012. http://data.ime.usp.br/sbbd2012/artigos/sbbd_min_01.html.

Desenvolvimento de um sistema de gestão para apoio à tomada de decisão no agronegócio da região do Alto Paranaíba

**Rodrigo Moreira¹, Dra. Adriana Zanella Martinhago¹,
Dr. Luis César Dias Drummond²**

¹Instituto de Ciências Exatas e Tecnológicas – ICET
Universidade Federal de Viçosa *campus* Rio Paranaíba (UFV)
Caixa Postal 22 – 38.810-000 – Rio Paranaíba – MG – Brasil

²Instituto de Ciências Agrárias – IAP
Universidade Federal de Viçosa *campus* Rio Paranaíba (UFV).

moreira_r@outlook.com, adriana.martinha@ufv.br, irriga@ufv.br

Abstract. *Organizations have computerized their processes through information systems, rural organizations have also followed this trend. With informatization, the amount of stored data has increased, and decision makers can support on information obtained from the data stored. The stored data is associated to the productivity of carrot and beet crops; and climate of the Rio Paranaíba region. These data are presented in forms of spreadsheets and are obtained from different sources, which makes the data analysis difficult. Therefore, the construction of a structured data repository is essential to enable the analysis, which aims to provide the information as a subsidy to decision makers in organizations.*

Resumo. *Organizações têm informatizado seus processos através de sistemas de informação, as organizações rurais também têm seguido essa tendência. Com a informatização, a quantidade de dados armazenados tem aumentado, e os tomadores de decisão podem apoiar em informações obtidas a partir dos dados armazenados. Os dados armazenados são associadas à produtividade das lavouras de cenoura e de beterraba; e clima da região de Rio Paranaíba. Estes dados são apresentados em formas de folhas de cálculo e são obtidos a partir de fontes diferentes, o que torna difícil a análise de dados. Portanto, a construção de um repositório de dados estruturado é essencial para permitir a análise, que tem como objetivo fornecer a informação como subsídio para os tomadores de decisão nas organizações.*

1. Introdução

Soluções de *Business Intelligence (BI)* ou Inteligência de Negócios, são cunhadas para padronização dos dados de forma adequada, em repositórios estruturados. A Inteligência de Negócios pode ser aplicada ao agronegócio, pois organizações desse setor tem investido em tecnologias de informação para diversos fins, como: monitorar o meio ambiente, automatização de processos, gerenciamento e etc. Cada novo sistema, fruto de investimentos em tecnologia da informação, são agentes criadores de dados, que posteriormente alimentarão ferramentas de *BI* [Barbieri 2001].

Criar um repositório de dados, *Data mart*, para armazenar dados relativos a produtividade de várias organizações do setor de agronegócio do Alto Paranaíba, juntamente com dados meteorológicos da região, proporcionará facilidade de acesso a pesquisadores, estudantes, professores e profissionais da área de agronomia às informações.

É importante possuir um repositório de dados irrepreensível, no que diz respeito a qualidade do seu conteúdo, portanto os dados devem ser tratados antes de compor do *Data mart*. Na fase de projeto do repositório de dados, é necessário atentar-se aos conceitos de modelagem multidimensional. Finalmente será possível apurar através de técnicas de *Business Intelligence*, níveis e a relação de produtividade contrastada com clima. A posse dessas informações viabiliza o processo decisório.

2. Data warehouse

Data warehouse (DW) é um banco de dados histórico, feito para o armazenamento dos dados extraídos de organizações. Para auxiliar os dados no apoio à tomada de decisão os dados devem ser armazenados em *DW*, eles são selecionados, integrados e organizados [Colaço 2004].

Segundo [Inmon et al. 1997] um *DW* deve possuir características como:

- não volatilidade: os dados em um *DW* só podem ser inseridos, atualizados e consultados;
- orientação à tópicos: *DW* armazena dados de um nicho específico, por exemplo: uma organização produtora de grãos, pode possuir um *DW*, onde este armazena, dados de clientes, informações sobre seus produtos e dados relativos ao clima;
- integrado: um *DW* deve ser capaz de armazenar dados que são oriundos de diversos sistemas transacionais da organização. Entretanto, é possível que cada sistema departamental utilize uma convenção para armazenar em seu banco de dados relacional esses dados operados diariamente. Quando esses dados são armazenados em *DW* pode haver incompatibilidades. Tome como exemplo o campo Estado (Unidade da Federação), em algum sistema pode-se utilizar a convenção de siglas, por exemplo: SP, ao passo que no *DW* pode ser usado um campo texto para o atributo em questão, isto é: São Paulo. O *DW* deve ser capaz de armazenar os dados fazendo uso de uma única convenção;
- varia com o passar do tempo: no *DW* os dados passam por atualizações no decorrer do tempo. Tome como exemplo um *DW* que continha dados de clientes solteiros no ano de 2009. Alguns anos depois, se parte ou todos esses clientes estiverem casados, o *DW* deve ser capaz de oferecer os dados para as duas circunstâncias;

2.1. Data mart

Data mart é um conjunto de dados flexíveis, baseados na maioria das vezes como dados atômicos, estes podem ser extraídos através de uma fonte operacional. São apresentados como modelo simétrico, resistente em ocasiões que o usuário faz consultas inesperadas. É uma fatia do *DW*, dedicado a manter dados de um único processo do negócio [Kimball and Merz 2000].

A Figura 1 ilustra um exemplo de um *Data mart*, tem-se Vendas, Clientes e Compras, cada um representa um processo de negócio específico [Laudon and Laudon 1999]. *Data marts* são específicos para um departamento da organização, portanto é uma

especialização do *DW*, feito para o armazenamento de dados em uma escala menor [Barbieri 2001].

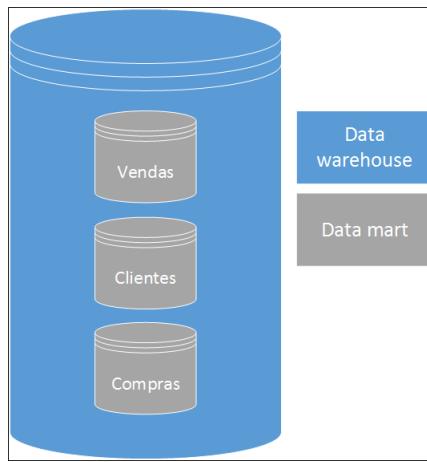


Figura 1. Abstração de um *Data warehouse*.

3. Extração, Transformação e Carga (ETC)

Segundo [Kimball and Merz 2000] pode-se determinar os objetivos do ETC como: (1) entregar dados de forma mais eficaz para as ferramentas de usuário final; (2) adicionar valor aos dados nas etapas de limpeza e padronização; (3) proteger e documentar a linhagem dos dados.

Extração, pode resumidamente ser conceituada como etapa de extração de dados provenientes dos sistemas transacionais. A etapa de transformação é responsável pela tarefa de padronização dos dados, para que a base de dados se torne inequívoca e completa. Carga, nessa etapa concentram-se atividades para o carregamento dos dados já tratados na estrutura multidimensional do armazém de dados [El-Sappagh et al. 2011].

4. Trabalhos Relacionados

- **Uso de Ferramentas de *Business Intelligence* na Análise de Desempenho de uma Empresa de Agronegócios**

[Lima and Boscarioli 2012] propõe a descrição de uma ferramenta para análise de desempenho em uma empresa, com o uso do processo de *BI*. Focada no agronegócio a empresa desenvolve atividades como comercialização de insumos e beneficiamento de grãos.

Para realização do trabalho foi investigada características da empresa para facilitar o entendimento dos processos internos e externos. Elaborou-se métricas de desempenho que são essenciais para análise corporativa do agronegócio. Fora arquitetado um *Data mart*, com foco na análise dos indicadores de desempenho. Também elaborou-se o agente ETC, para refinar a base de dados com o que realmente é importante e contribui positivamente com o processo de *BI*.

O trabalho descrito é relacionado com o desenvolvido na questão de criação de *Data mart*, para ser usado pela ferramenta de *BI*, para obtenção de informações que venham ser úteis na tomada de decisão. Possui divergência com o proposto, pois no trabalho

relacionado é criado um *Data mart* para armazenamento de dados de venda, no desenvolvido foi utilizado um *Data mart* para armazenar dados de clima e produção.

- **Representação de comercialização agropecuária através de modelo de *Data warehouse***

O trabalho de [Correa 2009] apresenta um estudo sobre modelos de dados dimensionais aplicados à definição de *DW* e *Data marts* com foco em agronegócio. É abordado questões sobre análise de informações e criação de *Data marts* para armazenar dados de pecuária e grãos. Para viabilizar a realização de processamento analítico online, utilizou-se como cadeias de comercialização pecuária e grãos como soja e milho. Justifica-se o uso dessas cadeias a importância no agronegócio brasileiro.

Com o desenvolvimento do trabalho foi possível fazer análises de mercado. Pôde ser feito cruzamento de informações, avaliar os impactos das variações de preços. O que apoia a tomada de decisões. O desenvolvimento de um *Data warehouse* possibilitou o armazenamento centralizado de dados sobre comercialização e produção.

Com relação ao trabalho desenvolvido, esse se relaciona no quesito de modelagem de armazém de dados históricos com foco no agronegócio. E no uso do Sistema de Gerenciamento Bando de Dados (SGBD) para gerenciamento dos dados armazenados.

5. Métodos

Baseado na Figura 2 é apresentado os desdobramentos das etapas que compõe o ciclo de desenvolvimento do trabalho proposto.

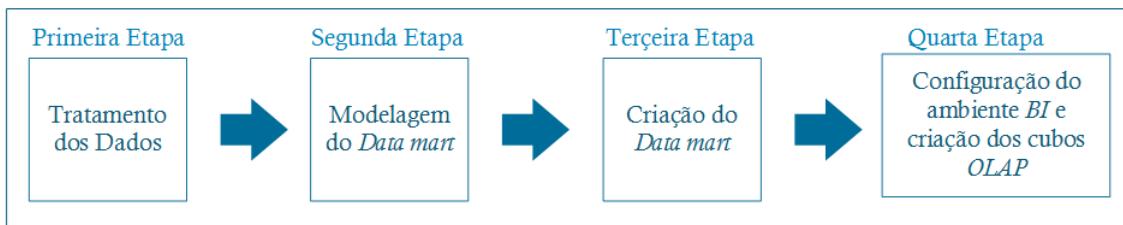


Figura 2. Etapas do desenvolvimento do Trabalho.

A primeira etapa representada na Figura 2 contempla a atividade de tratamento dos dados. Eles foram obtidos da Estação Meteorológica instalada no *Campus I UFV Rio Paranaíba* e foram exportados em forma de planilhas eletrônicas. O sistema da Estação Meteorológica é programado para coletar dados a cada 60 min, esses procedimentos são realizados na frequência mencionada desde janeiro de 2009. Os dados contidos nas planilhas eletrônicas estão dispostos em forma de linhas, onde cada linha representa um registro. Cada registro contém: Data, Hora, Temperatura Máxima, Temperatura Mínima, Velocidade do Vento, Direção do Vento, Humidade Relativa do Ar, Evapotranspiração. Também foram tratados dados quantitativos da produtividade de culturas de cenoura e beterraba de fazendas da zona rural dos municípios de Rio Paranaíba – MG e São Gotardo – MG, cedidos pelo Grupo *Sekita Agronegócios*.

A etapa de tratamento dos dados consiste em retirar campos nulos causados por eventuais falhas técnicas ou não preenchimento, no caso das tabelas de produtividade. A relevância do tratamento é justificada pela necessidade de armazenar dados íntegros e

confiáveis no *Data mart*. Para o tratamento foi utilizada a ferramenta *Data Integration* da suíte *Pentaho*. Pois permite-se escolher registros e campos da planilha eletrônica a serem carregados pelo *Data mart*, também ignorar campos nulos e aqueles que não compõe o *Data mart*, seja na tabela de Fatos ou em tabelas Dimensão.

A segunda etapa contém a atividade de modelar o *Data mart*. Para modelagem foi utilizado a ferramenta *open source MySQL Workbench*. Na modelagem foi adotado a abordagem *Star Schema*, para que o banco de dados seja consolidado na forma multidimensional como propõe [Colaço 2004].

Os benefícios de um *Star Schema* ultrapassa os de um modelo normalizado, pelo fato das consultas no armazém de dados ser apenas de leitura [Teorey et al. 2014]. Em adição, [Weininger 2002] propõe técnicas para melhorar as consultas em modelos *Star Schema*, que podem ser exploradas por duas vertentes; (1) *hash push down*, (2) *bit-vector push down*. Ambas objetivam a redução de linhas, ao realizar operação do tipo *Join*.

A terceira etapa consiste em criar e povoar o *Data mart* com base na modelagem desenvolvida. Foi utilizado o Sistema Gerenciador de Banco de Dados (SGBD) *PostgreSQL*, pelo seu bom desempenho, por ser *open source* e apresentar boa compatibilidade com a suíte *Pentaho*, principalmente no que diz respeito a *plugins*¹ de conexão com os módulos do *Pentaho* [Pires et al. 2006].

Após consolidado o *Data mart* e com os dados carregados, segue as atividades da quarta etapa. Nesta etapa foi configurado o ambiente *Pentaho Business Analytics* para possibilitar análises. Também foi criado cubos *OLAP*² como parte da solução de *BI*, com suas características relacionadas a granularidade e métricas. Para criação dos cubos foi utilizado o módulo *Mondrian* da suíte *Pentaho*. Após criado cubos *OLAP*, eles puderam ser publicados para o *Pentaho BI Server* para que fosse possível visualizá-los e analisá-los, seja de forma gráfica ou tabular.

6. Resultados e Discussões

Após seguir sistematicamente a metodologia, foi desenvolvido o modelo lógico do *Data mart* que pode ser visualizado na Figura 3. Para tal, foi utilizado a ferramenta *MySQL Workbench*. No modelo, pode ser visualizado a tabela de fatos; “tableFato” como centro do esquema e, as demais tabelas: “tableData”, “tableClima”, “tableGleba”, “tableCultura” e “tableFazenda”. Todos atributos da tabela foram criados para armazenar valores de campos provenientes das planilhas de clima e produção. A tabela de fatos possui a chave estrangeira (*Foreign Key*) de cada dimensão que a habilita contatar qualquer tabela na borda do esquema, possui também um atributo de medida, conceitualmente denominado como métrica; no caso do trabalho desenvolvido é a produtividade. As demais tabelas, são tabelas de dimensão, elas possuem seus atributos e uma chave primária (*Primary Key*) para poder ser ligada a tabela de fatos, dessa forma é possível apurar a produtividade por alguma dimensão.

¹Também conhecido como módulo de extensão, utilizado para adicionar função a outros programas maiores.

²*On-line Analytical Processing (OLAP)* ou Processamento Analítico On-Line é conjunto de técnicas usadas para tratar informações contidas em repositório de dados estruturados *DW* ou *Data mart* [Colaço 2004]

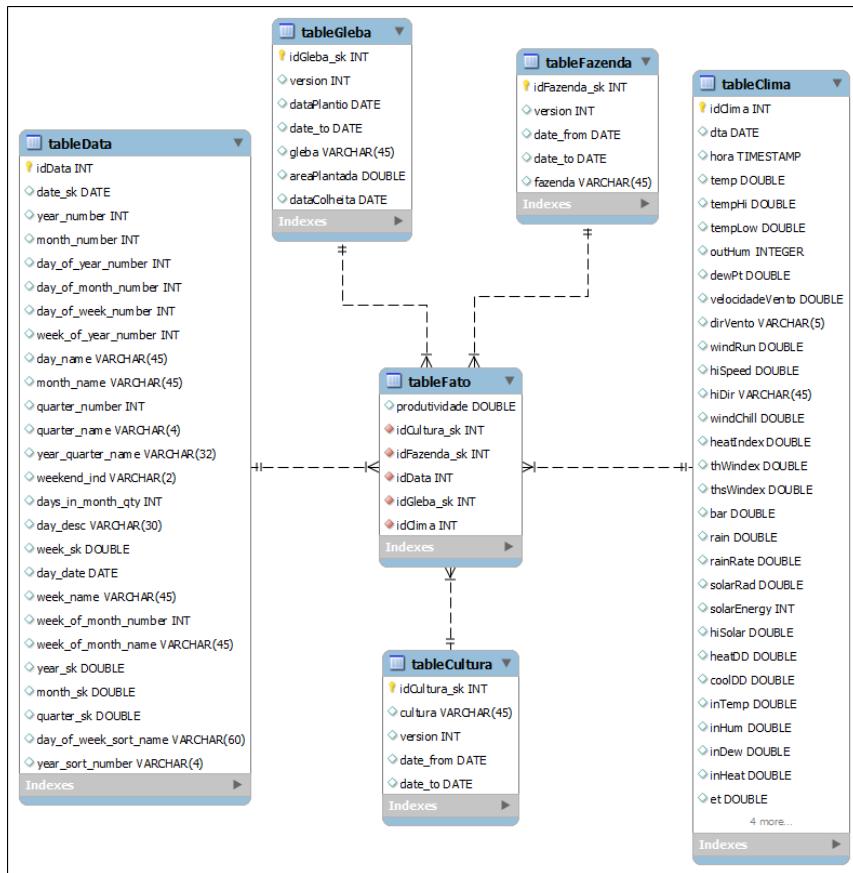


Figura 3. Modelo lógico do *Data mart*.

A Figura 4 ilustra o processo de Extração, Transformação e Carga (ETC) modelado para armazenar os dados na tabela de fatos do *Data mart*. O primeiro componente foi configurado para realizar a leitura linha a linha da planilha de produção. Na medida que são lidas elas seguem para o componente “FiltroLinhasNulas” que trata a eventualidade de campos nulos, se houver a linha é descartada pelo componente “DescartaLinhas”. Quando a linha lida é íntegra, ela segue para o componente “DimensionamentoCultura”, esse componente foi configurado para comparar o nome da cultura da planilha com o nome da cultura na “tableCultura”, quando for igual esse componente encaminha o “idCultura” para o próximo componente.

O componente “DimensionamentoGleba” foi configurado para comparar o nome da Gleba registrado na planilha com o nome da Gleba armazenado na “tableGleba”, quando for igual esse componente encaminha o “idGleba” para o próximo componente. O componente “DimensionamentoFazenda” segue a mesma sistemática, comparara o nome da Fazenda registrado na planilha com o nome da Fazenda armazenado na “tableFazenda”, quando for igual esse componente encaminha o “idFazenda” para o próximo.

O componente “DimensionamentoData” foi configurado para vincular uma data da “tableData” com a data de plantio da cultura. O componente foi configurado para comparar se a data da tabela é igual a alguma data de plantio registrada na planilha de produção, quando houver, o componente encaminha para o próximo o “idData”. No próximo componente; “DimensionamentoClima”, foi configurado para comparar se a data

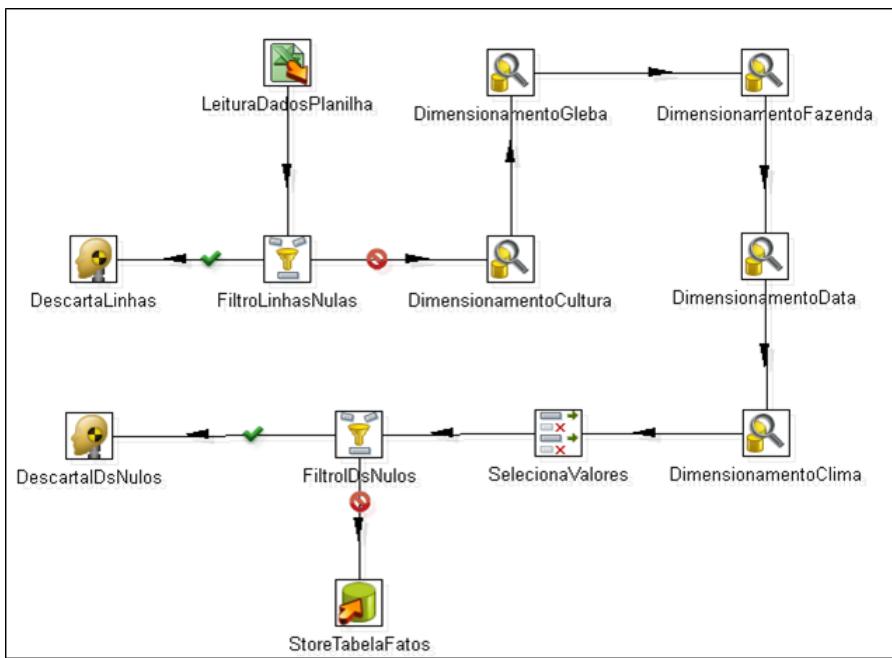


Figura 4. Transformação da tabela Fato.

do clima contida na tabela “tableClima” é igual a data de plantio registrada na planilha de produção, quando forem iguais, o componente encaminha o “idClima”.

Foi criado um componente “SelecionaValores”, esse componente é responsável por selecionar os campos que serão armazenados na tabela de fatos. O componente seleciona: “produtividade”, “idFazenda”, “idClima”, “idGleba”, “idData” e “idCultura” para poder compor a tabela de fatos do *Data mart*. Após selecionado os campos, o componente “FiltroIDsNulos” verifica a ocorrência de “ids” nulos. A ocorrência de algum, faz com que o registro dimensionado seja descartado, para assegurar as restrições de integridade de chave no *Data mart*.

Na ocorrência de “ids” íntegros encaminha-se o fluxo de dados para o último componente, “StoreTabelaFatos”. Nesse componente foi elaborado consulta *SQL* para fazer a inserção na “tableFato”. O registro armazenado na tabela de fato consegue através das chaves estrangeiras (*Foreign Key*) vincular uma produtividade; métrica escolhida com: Fazenda, Gleba, Cultura, Data e Clima. Com essa estrutura do *Data mart* pode-se aplicar técnicas de consultas *OLAP* através de ferramentas *BI*.

Com o desenvolvimento do sistema é possível apurar pela ferramenta *Pentaho Business Analytics* a produtividade das culturas de cenoura e beterraba da região do Alto Paranaíba de forma tabular ou gráfica. De maneira semelhante é possível visualizar uma condição climática e a produtividade de uma determinada época. Esse tipo de informação é essencial para os gestores de organizações rurais, pois os possibilitam avaliar investimentos ao plantar determinada cultura em uma época específica. As informações que um sistema de *BI* proporciona aos tomadores de decisão são subsídios somadores para uma decisão acertada.

As ilustrações subsequentes são cópias autênticas dos gráficos gerados pelo *Pentaho Business Analytics*. A Figura 5 ilustra o nome das fazendas e o total de produção

da cultura de cenoura. A fazenda “Lote46” apresenta maior produção dessa cultura. A Figura 6 ilustra de forma gráfica a produtividade de cenoura e a condição climática que a cultura foi submetida desde o plantio até a colheita. A Figura 7 ilustra o gráfico da produtividade em caixas por hectare da cultura de cenoura. Os detalhes da barra contidos na caixa preta retrata que, em 15.8 hectare de área plantada a produção total foi de 6.172 caixas por hectare.

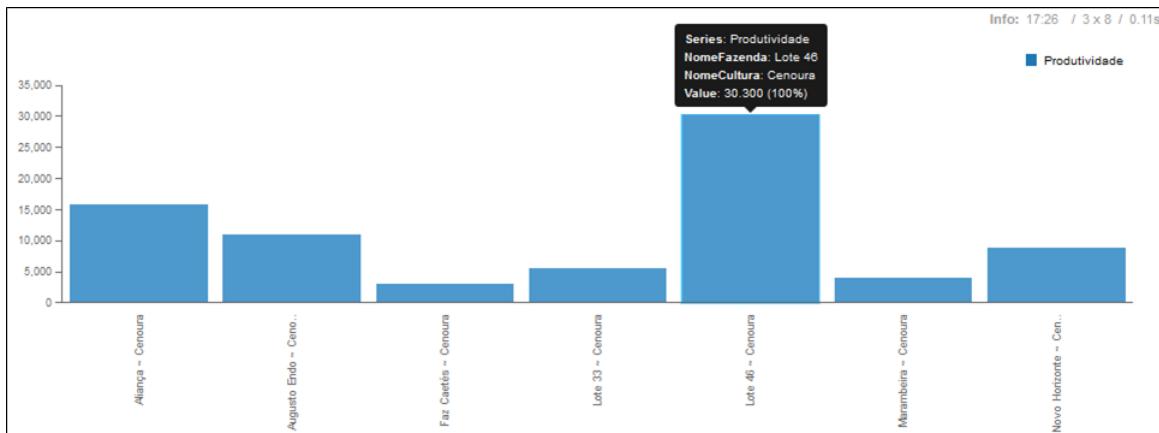


Figura 5. Produtividade: Fazenda × Cultura

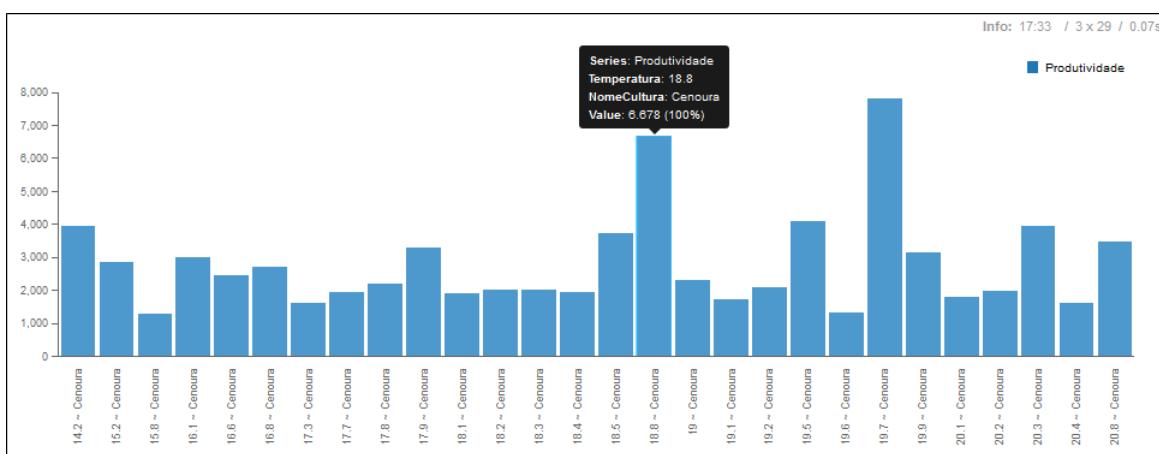


Figura 6. Produtividade: Cultura × Temperatura

O gráfico ilustrado na Figura 8 mostra a distribuição média do clima da região do Alto Paranaíba dos anos de 2008 à 2011. Os detalhes de um ponto específico podem ser vistos na caixa preta, onde é retratado que na data 23-04-2010 a temperatura média foi de 23,605 °C.

7. Conclusões

Foi possível analisar a produtividade que é dada em caixas por hectare por diversas dimensões. A possibilidade de criar gráficos com o arrastar de componentes mostrou a potencialidade da análise multidimensional de dados para apoio à decisão. A geração de gráficos de clima possibilita adquirir informações sobre vários aspectos que compõe uma

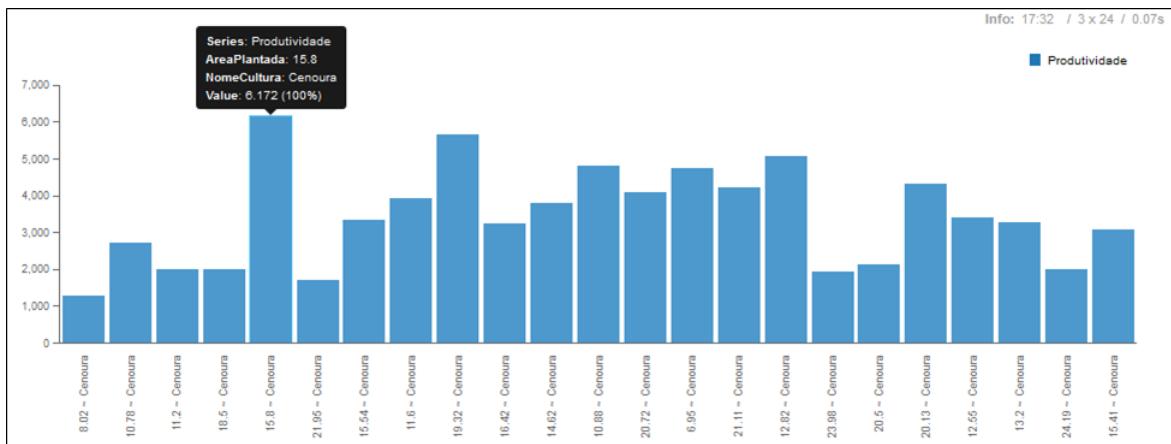
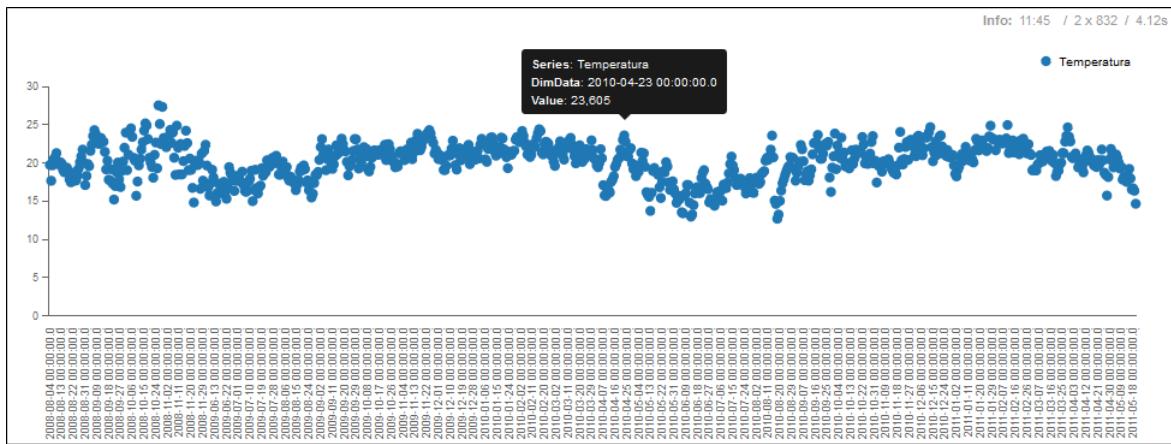


Figura 7. Gráfico Produtividade Cultura × Área Plantada.



Referências

- Barbieri, C. (2001). *BI-business intelligence: modelagem e tecnologia*. Axcel Books.
- Colaço, M. (2004). *Projetando Sistemas de Apoio à Decisão Baseados Em Data Warehouse*. AXCEL BOOKS, 1st edition.
- Correa, F. E. (2009). Representação de comercialização agropecuária através de modelo de data warehouse.
- El-Sappagh, S. H. A., Hendawi, A. M. A., and Bastawissy, A. H. E. (2011). A proposed model for data warehouse {ETL} processes. *Journal of King Saud University - Computer and Information Sciences*, 23(2):91 – 104.
- Inmon, W. H., Zachman, J. A., and Geiger, J. G. (1997). *Data Stores, Data Warehousing and the Zachman Framework: Managing Enterprise Knowledge*. McGraw-Hill, Inc., New York, NY, USA, 1st edition.
- Kimball, R. and Merz, R. (2000). *Data webhouse: construindo o data warehouse para a Web*. Campus.
- Laudon, K. C. and Laudon, J. P. (1999). *Sistemas de informação: com Internet*. LTC Editora.
- Lima, V. M. d. and Boscarioli, C. (2012). Uso de ferramentas de business intelligence na análise de desempenho de uma empresa de agronegócios. pages 420–431.
- Pires, C. E. S., Nascimento, R. O., and Salgado, A. C. (2006). Comparativo de desempenho entre banco de dados de código aberto. *Escola Regional de Banco de Dados, Anais da ERBD 06*.
- Teorey, T., Lightstone, S., Nadeau, T., and Jagadish, H. V. (2014). *Projeto e Modelagem de Banco de Dados*. Rio de Janeiro, RJ, Brazil, 2nd edition.
- Weininger, A. (2002). Efficient execution of joins in a star schema. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, pages 542–545, New York, NY, USA. ACM.

Integração de Dados Biológicos no Programa Net2HomologyWeb

Daniel Orlandi Mincato¹, Daniel Luis Notari¹

¹ Centro de Ciências Exatas e da Tecnologia – Universidade de Caxias do Sul (UCS)

Rua Francisco Getúlio Vargas, 1130 - CEP 95070-560 - Caxias do Sul – RS – Brazil

domincat@ucs.br, dlnotari@ucs.br

Abstract. A scientific workflow entitled *Net2Homology* has been designed to appraise similarity among PPI (Protein-Protein Interaction) networks from two different species and it establishes physical and functional frameworks among them. The *Net2Homology* also performs the data correlation among the NCBI and STRING's database and one of its issues is the act of linking the protein identifiers derived from the PSI-BLAST and NCBI. We have used a local database to integrate this information to a new *Net2Homology* version on the web platform for the execution of the consultations of the protein-protein network interactions in the STRING's database. This article proposes a local database method to process, integrate and store proteins information on a new web version of this program.

Resumo. O programa *Net2Homology* é um workflow científico que realiza comparações entre redes de interação proteína-proteína, de duas espécies diferentes, identificando as estruturas físicas e/ou funcionais semelhantes entre elas. Para isso a ferramenta realiza a integração de dados entre os bancos de dados do NCBI e o banco de dados STRING. Um problema deste programa consiste em vincular os identificadores de proteínas resultados do programa PSI-BLAST, do NCBI, para a realização das consultas das redes de interação proteína-proteína no banco de dados STRING. Foi então realizada a utilização de um banco de dados local (PostgreSQL) para integrar essas informações em uma nova versão da ferramenta *Net2Homology* na plataforma web.

1. Introdução

Banco de dados é um repositório de informações que representam aspectos do mundo real, possuem coerência lógica entre si e inferem significado e interesse para um grupo de usuários (Elmasri e Navathe, 2011). Dados biológicos são as unidades básicas de informação que representam fenômenos concretos ou abstratos associados a um contexto (Puga, França e Goya, 2014), sendo que este contexto, neste caso, está contido na área da biologia.

DNA e RNA são o material genético que armazenam as informações de um indivíduo e seu plano de desenvolvimento. Suas sequências são longas (na ordem de milhares mesmo para os seres vivos mais simples) e lineares (Lesk, 2008). Proteínas são moléculas que determinam a estrutura e atividade dos organismos, como nossos cabelos ou nossos anticorpos (LESK, 2008). Redes de interação entre proteínas são estruturas formadas pela conexão física e/ou funcional das proteínas que a formam. Um banco de dados biológico pode armazenar diferentes tipos de dados, dos diferentes níveis da

biologia. Esses bancos também devem fornecer meios de consulta eficientes para a coleta de informações específicas (Barnes e Gray, 2003; Jones e Pevzner, 2004; Lesk, 2008). A análise de homologia é o método onde se analisa a similaridade entre duas sequências genéticas de qualquer tamanho, resultando em um conceito que deve considerar a sensibilidade e também a seletividade (LESK, 2008). Esses métodos são aplicados nas ferramentas de *Basic Local Alignment Search Tool* (BLAST) que retornam sequências similares a uma sequência passada como dados de entrada nessas ferramentas.

Um *workflow* diz respeito à automatização de procedimentos, onde documentos, informações ou tarefas são passadas entre os participantes de acordo com um conjunto pré-definido de regras, para se alcançar ou contribuir no objetivo global de um negócio (Wfmc, 2004; Sommerville, 2007). São usados para descrever experimentos *in silico*¹ em bioinformática (Silva et al, 2010; Chirigati e Freire, 2012) e, o seu uso tem crescido muito nos últimos anos (Silva et al, 2010; Linke et al., 2011; Chirigati e Freire, 2012). Apesar de um *workflow* permitir a organização manual de procedimentos, na prática a maioria dos *workflows* são organizados dentro de um contexto do sistema de informação para prover um apoio automatizado aos procedimentos (Wfmc, 2004; Sommerville, 2007).

A ferramenta Net2Homology foi desenvolvida como um *workflow* científico que tem como objetivo realizar a comparação entre duas redes de interação proteína-proteína, de dois organismos diferentes, mostrando as estruturas e funções semelhantes entre elas (Notari, 2012). Um problema na sua versão *desktop* está na integração de dados entre os dados extraídos dos bancos de dados do NCBI pela da execução do programa PSI-BLAST, com os dados que devem ser utilizados para a realização das consultas das redes de interação proteína-proteína no banco de dados STRING.

Neste artigo é apresentada uma proposta de desenvolvimento de uma versão para o programa Net2Homology na plataforma *web*, visando também aprimorar a qualidade dos dados extraídos dos bancos de dados do NCBI a fim de melhorar a busca das informações das respectivas proteínas no banco de dados STRING. Para esta integração, foi proposta a utilização de um banco de dados local para armazenamento das referências diretas entre os identificadores das proteínas dos bancos de dados do NCBI e os identificadores das proteínas do banco de dados STRING.

A seção 2 apresenta informações sobre os dados e métodos biológicos utilizados. Na seção 3 é apresentado a versão atual do programa Net2Homology, trabalhos relacionados e os problemas desta versão. A seção 4 aborda a proposta de solução, detalhes de implementação, um estudo de caso e uma análise dos resultados obtidos. Por fim, na seção 5 são apresentadas as conclusões e trabalhos futuros.

2. Dados Biológicos

O Genbank² é um banco de dados de sequências de DNA e uma coleção de anotações de todas as sequências públicas de DNA de diferentes organismos, desenvolvido e distribuído publicamente pelo National Center of Biotechnology Information (NCBI) (Jones e Pevzner, 2004; Lesk, 2008). O GenBank tem sido expandido para incluir dados

¹ *In silico*: realização de alguma tarefa por meio da utilização de um computador ou de um processo computacional.

² Pode ser acessado em <http://www.ncbi.nlm.nih.gov/genbank/>.

de expressão de sequências, dados de sequências de proteínas, estrutura tridimensional de proteína, taxonomia, informações sobre diferentes tipos de interações biológicas e literatura biomédica (Benson et al, 2014). Além de bancos de dados, o NCBI fornece serviços e ferramentas para analisar os dados, sendo uma destas ferramentas o programa BLAST (Basic Local Alignment Search Tool).

O BLAST³ usa métodos de similaridade para comparação e alinhamento de sequências de nucleotídeos ou de proteínas (Altschul, 1990; Altschul, 1997). O resultado é baseado em cálculos estatísticos após a obtenção de casamentos (*matches*) significativos entre as sequências (Altschul, 1990; Altschul, 1997). O BLAST pode ser usado para inferir relacionamentos funcionais e evolucionários entre sequências, bem como identificar membros de uma família de genes (Altschul, 1990; Altschul, 1997). O Psi-Blast (Position-Specific Iterated BLAST) é um dos programas disponíveis para encontrar proteínas distamente relacionadas ou encontrar novos membros de famílias de proteínas (Gish e States, 1993). Através do alinhamento de sequências podem-se identificar que duas sequências são homólogas se elas possuem uma sequência ancestral comum (Barnes e Gray, 2003; Lesk, 2008). O *e-value* representa o número esperado de alinhamentos locais distintos em um casamento (*match*) realizado para a comparação de duas sequências aleatórias de quaisquer tamanhos (Altschul, 1990; Altschul, 1997). Em outras palavras, o valor do *e-value* descreve o nível de ruído (*hit*) do alinhamento podendo variar entre zero e o número de sequências procuradas no banco de dados (Barnes e Gray, 2003; Lesk, 2008).

Outro banco de dados utilizado para consultar informações sobre proteínas é o STRING⁴ (Search Tool for the Retrieval of Interacting Genes/Proteins) que é uma ferramenta para realizar metabuscas com o intuito de se obter redes de interação de proteínas (Jensen et al, 2009). O STRING disponibiliza um serviço que fornece informações sobre associações funcionais entre proteínas em um único lugar, facilitando a consulta e navegação nos dados em interfaces interativas e intuitivas (Szklarczyk et al, 2011). Um dos métodos usados para trabalhar com o banco de dados STRING é a Biologia de Sistemas. Esta é uma área interdisciplinar com o objetivo de compreender como funcionam as interações entre populações de moléculas, células e organismos devido ao aumento da complexidade de processos biológicos (Karsenti, 2012).

3. Workflow Científico Net2Homology

A aplicação Net2Homology⁵ foi desenvolvida como um *workflow* científico com o objetivo de realizar a comparação entre duas redes de Interação Proteína-Proteína (PPI), de dois indivíduos diferentes, verificando as estruturas e funcionalidades conservadas durante o processo evolutivo através de uma análise de homologia com o programa BLAST (NOTARI, 2012). As etapas desse *workflow* estão ilustradas na Figura 1, possuindo cinco etapas principais: *i*) seleção de proteínas e organismo origem (passos A e B); *ii*) seleção de organismo alvo, configuração e execução do BLAST (C a H); *iii*) processamento do resultado do BLAST, seleção de dados e consulta das redes de interação

³ Pode ser acessado em <http://blast.ncbi.nlm.nih.gov/Blast.cgi>.

⁴ Pode ser acessado em <http://string-db.org/>.

⁵ Maiores informações podem ser obtidas em <http://www.danlian.com.br/homology/>.

proteína-proteína (passos I a N); iv) comparação das redes de interação de proteínas geradas (passos O a Q); e, v) visualização dos resultados finais (passos R a T).

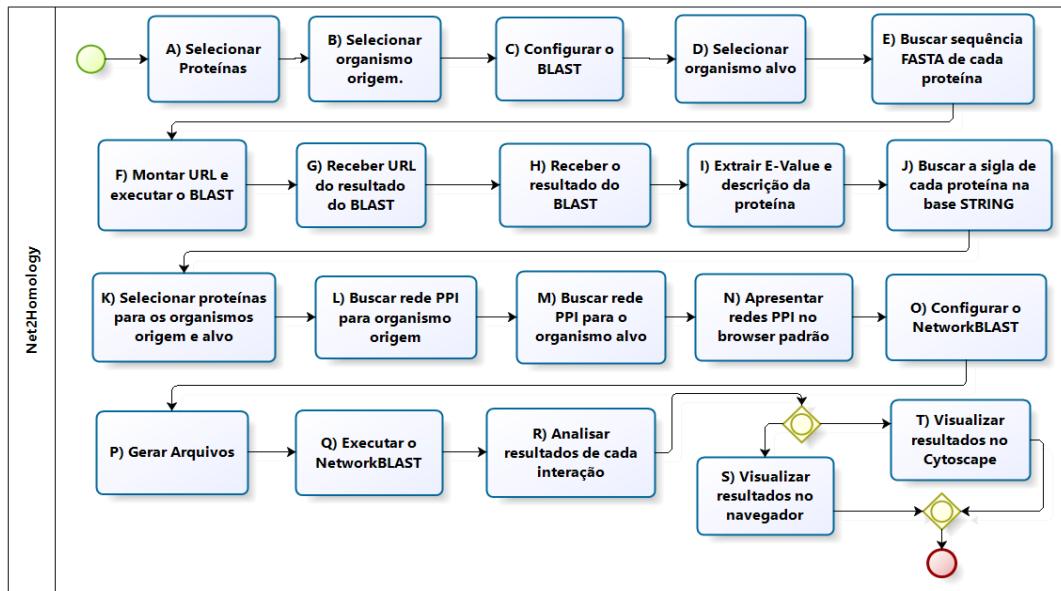


Figura 1 - Fluxo de execução do workflow Net2Homology.

O maior problema encontrado está na consulta dos identificadores das proteínas no banco de dados STRING, feita antes de consultar as redes de interação. Devido ao fato da aplicação realizar processamento de texto sobre o resultado do programa BLAST para encontrar dados da proteína, nem sempre encontra dados relevantes. Devido a isso, a aplicação não consegue encontrar o identificador da proteína no banco de dados STRING em alguns casos e, por consequência, não consegue buscar a rede de interação proteína-proteína. Os identificadores de proteínas utilizados pelos bancos de dados GenBank e STRING são diferentes. Por isto, realizar um casamento puro e simples não é viável.

3.1 Trabalhos Relacionados

O BioExtract é uma plataforma, disponibilizada na web, com o propósito de auxiliar os pesquisadores na análise de dados genéticos, oferecendo recursos para o desenvolvimento de workflows de bioinformática possuindo uma interface flexível para a execução de consultas aos bancos de dados de proteínas e de nucleotídeos não-redundantes do NCBI (Lushbough, Gnimieba e Dooley, 2013). Através dele é possível filtrar os resultados obtidos das consultas nesses bancos de dados e enviá-los como dados de entrada para ferramentas de análise, além de permitir salvar os dados extraídos, integrando-os na ferramenta e utilizando-os como conjunto de dados pesquisáveis (Lushbough, Gnimieba e Dooley, 2013).

O SEMEDA é uma ferramenta criada com o propósito de gerar um repositório de metadados e anotações sobre os bancos de dados biológicos, além de criar e armazenar ontologias utilizando vocabulários controlados (Köhler, Philippie e Lange, 2003). As ontologias representam anotações e, através delas, é possível vincular os metadados para integrar as informações dos bancos de dados envolvidos (Köhler, Philippie e Lange, 2003).

A Tabela 1 apresenta uma comparação entre as duas ferramentas e o programa Net2Homology analisando as suas finalidades, métodos de integração, entre outras informações. O Net2Homology foi desenvolvido como um *workflow* científico enquanto que o BioExtract permite criar workflows científicos. O SEMEDA permite criar um vocabulário comum com o uso de ontologias. As três ferramentas permitem consultar os bancos de dados do NCBI. Tanto o Net2Homology quanto o BioExtract permitem realizar o casamento dos dados biológicos. Outras informações relevantes destas ferramentas é a forma como é feita a busca dos dados, ou por arquivos ou por acesso via *web services*. Somente o Net2Homology é uma ferramenta *desktop*.

Tabela 1 - Comparaçāo entre as ferramentas.

	Net2Homology	BioExtract	SEMEDA
Tipo de Ferramenta	<i>Workflow</i>	Plataforma	Integrador
Bancos de Dados Integrados	NCBI X STRING	NCBI	Qualquer
Onde Integra	Internamente	NCBI	Internamente
Método de Integração	Cruzamento	Cruzamento	Ontologia e Semântica
Interface de Integração	Arquivos texto	Interconexões do NCBI	Interconexões
Disponível na Web	Não	Sim	Sim

4. Workflow Científico Net2HomologyWeb

O objetivo principal desse trabalho é resolver o problema de integração entre os bancos de dados do NCBI e o banco de dados STRING, aprimorando o algoritmo que realiza essa integração. Outro objetivo é realizar a migração de plataforma da ferramenta, tornando-a uma aplicação na *web*. Para essa migração foi decidido pela utilização da estrutura lógica Model-View-Controller (MVC) devido à utilização do framework JSF que é baseado nessa arquitetura. Esta seção apresenta a proposta de solução e os resultados obtidos.

4.1 Integração entre os bancos de dados do NCBI e STRING

O procedimento que busca os identificadores das proteínas selecionadas no banco de dados STRING precisou ser aprimorado, a fim de diminuir a quantidade de redes de interação que não podem ser consultadas por problemas nessa etapa. Para esse aprimoramento, foram necessárias as seguintes funcionalidades: *i*) armazenar a relação (identificador da espécie, identificador do NCBI, identificador do STRING) em banco de dados local; *ii*) consultar o identificador do STRING utilizando a descrição da proteína; e, *iii*) consultar o NCBI para buscar mais informações que identifiquem a proteína a ser consultada no STRING.

Esse procedimento de consulta acontece entre os passos de consulta ao STRING e visualização dos resultados do STRING. O fluxo de atividades deste procedimento está ilustrado na Figura 2.

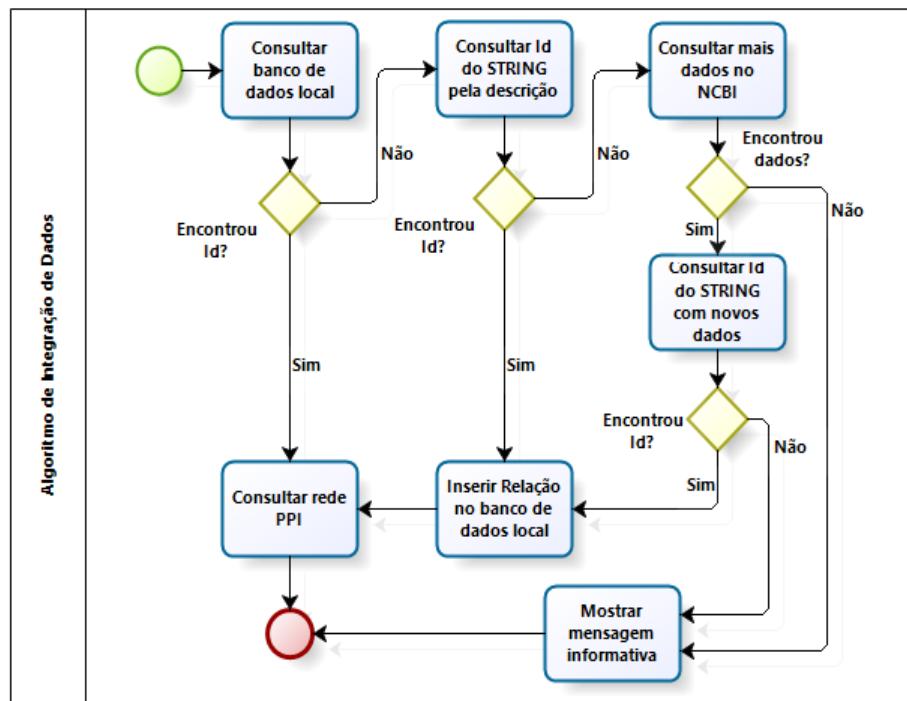


Figura 2 - Fluxo do algoritmo de busca do identificador no banco de dados STRING.

A integração de dados proposta foi feita através da criação de um banco de dados local (PostgreSQL), onde são armazenadas as referências diretas entre os identificadores das proteínas referente às duas fontes de dados. O diagrama entidade-relacionamento para a integração dos dados está ilustrado na Figura 3. Nesta base de dados são armazenadas as informações dos organismos (espécies biológicas) provenientes de um arquivo⁶ em formato TSV com as referências diretas entre o banco de dados STRING e os bancos de dados do NCBI, das proteínas envolvidas e do banco de dados na qual foi feita a pesquisa.

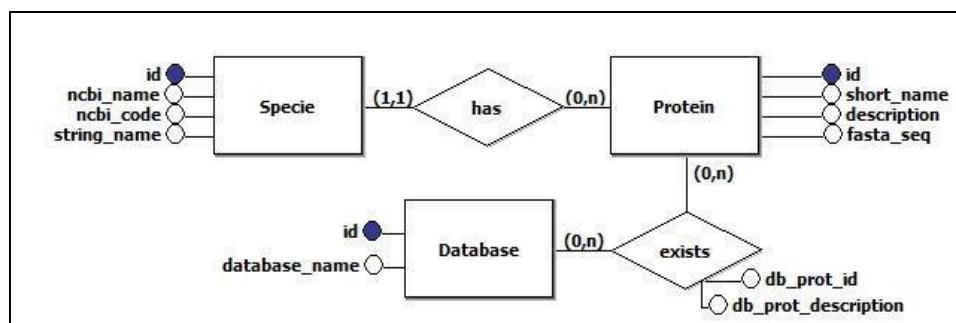


Figura 3 - Diagrama entidade-relacionamento da aplicação Net2HomologyWeb.

⁶ Os dados estão em formato TSV e podem ser acessados em http://stringdb.org/newstring_download/species.v9.1.txt.

4.3 Estudo de Caso

Para testar a nova versão da ferramenta⁷, na plataforma web, selecionou-se o organismo *Homo sapiens* e a proteína XPD. Após informar o organismo origem e as proteínas para consulta, é necessário informar o organismo alvo e configurar o BLAST informando o banco de dados, o algoritmo e o *e-value* máximo tolerável. Para a execução do estudo de caso, foi selecionado o organismo *Mus musculus*, o banco de dados GenBank, o algoritmo *blastp* (protein-protein BLAST) e o *e-value* máximo 0.005.

Após o término da execução do BLAST, é necessário realizar a seleção das proteínas da primeira etapa do *workflow* e dos resultados do BLAST recebidos para elas para realizar a consulta das redes PPI. Na execução do estudo de caso, foi selecionado o resultado do BLAST da proteína XPE com o melhor resultado, que obteve uma pontuação de 3885 na comparação com a sequência original. Ao término da consulta das redes PPI no banco de dados STRING, as imagens das redes PPI são apresentadas na última tela do *workflow* Net2Homology que foi abordada nesse trabalho. Em relação ao estudo de caso realizado e demonstrado, a tela para visualização das imagens das redes PPI está ilustrada na Figura 4.

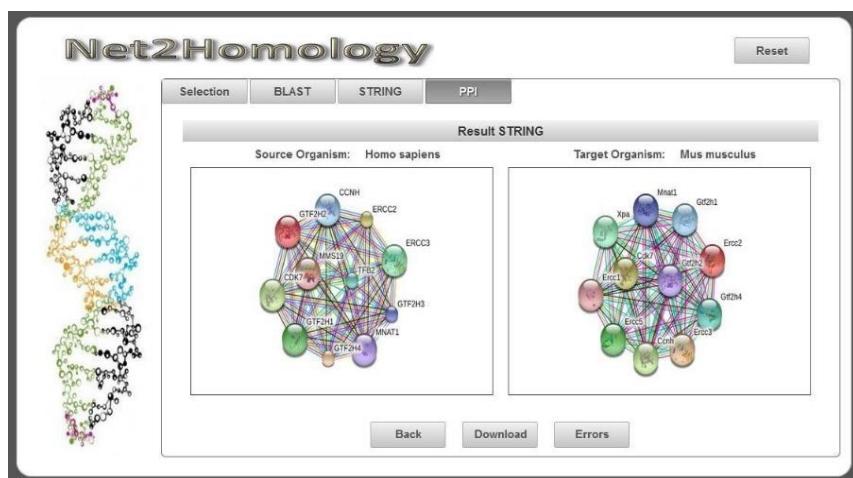


Figura 4 – Resultado da consulta ao banco de dados STRING.

Comparando os resultados obtidos na execução do programa Net2Homology na plataforma web, com a execução, para os mesmos dados de entrada, na plataforma desktop, é possível observar que os resultados obtidos foram semelhantes, conforme apresentado na Figura 5, onde temos destacado em A e C, as redes PPI obtidas pela execução da ferramenta na plataforma desktop, e em B e D as redes PPI obtidas pela execução da ferramenta na plataforma web.

⁷ Por questões do espaço disponível neste artigo não foram mostradas todas as telas de cada passo do workflow para a execução do programa Net2Homology. A execução completa pode ser acessada em (Mincato, 2014) a ser disponibilizada na web até março de 2015.

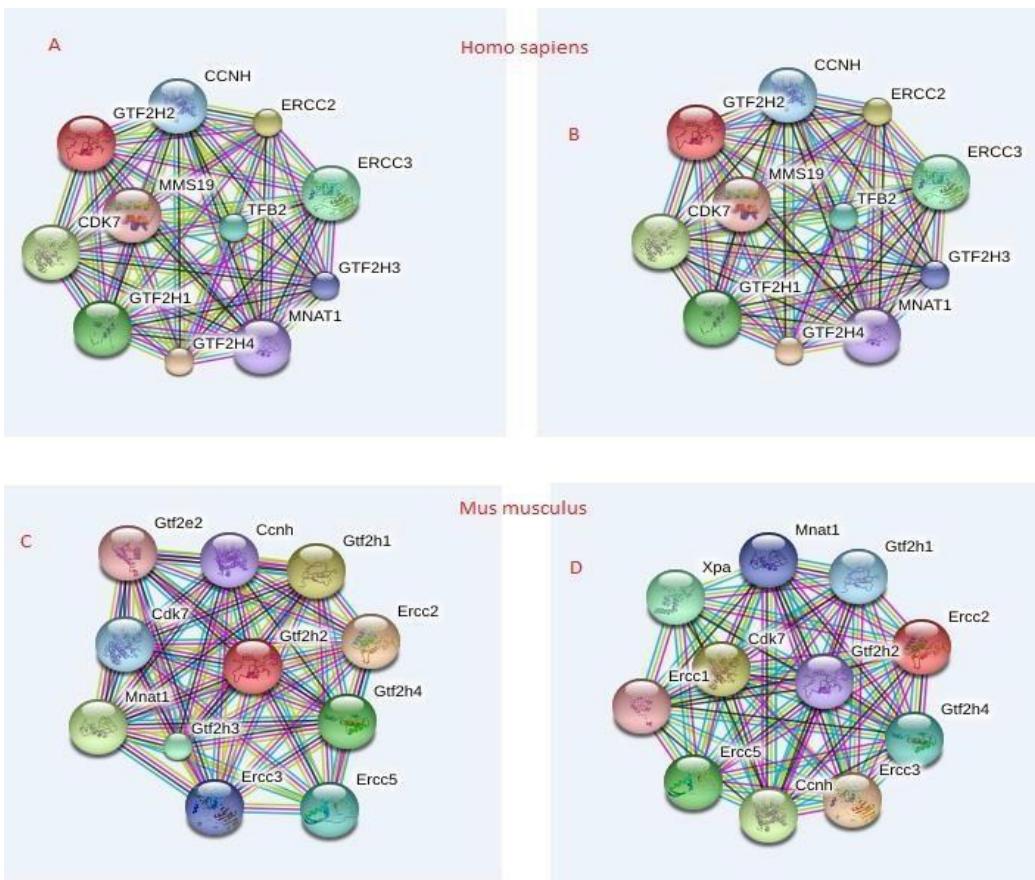


Figura 5 - Comparaçao dos resultados obtidos das execuções *desktop* e *web*.

Uma das etapas de consulta das redes PPI no banco de dados STRING é a obtenção dos identificadores das proteínas selecionadas durante a execução do workflow. Esses identificadores provêm do banco de dados STRING e são necessários para a consulta das redes PPI. Nesse estudo de caso, a proteína XPD não existia no banco de dados local, por nunca ter sido consultada anteriormente. A avaliação da alteração do algoritmo de busca dos identificadores no banco de dados STRING foi feita mediante análise dos dados que constam no banco de dados local. Com base nisso, foi realizada uma consulta nesse banco de dados local, a fim de buscar os dados que o mesmo possui a respeito da proteína XPD.

Nessa consulta ao banco de dados local, foram retornados quatro registros, dois referentes ao organismo *Homo sapiens* e os outros dois referentes ao organismo *Mus musculus*. Para os registros de um mesmo organismo, foi identificado que havia sido armazenado o identificador da proteína XPD consultado no NCBI, assim como seu identificador no banco de dados STRING. Dessa forma, foi identificado que o algoritmo aprimorado está armazenando as relações entre os bancos de dados do NCBI e o banco de dados STRING, quando as encontra. Portanto, quando consultada novamente essa mesma proteína, o seu identificador no banco de dados STRING será retornado do banco de dados local, não sendo mais necessário fazer consultas em bancos de dados remotos para a obtenção dessa informação.

5. Conclusão

A ferramenta Net2Homology foi aprimorada, executando agora na plataforma *web*. Referente a esse objetivo, só não foi completamente atingido porque não foi possível disponibilizar a ferramenta no servidor de hospedagem por causa de falhas que o mesmo apresentou. Esses problemas não foram solucionados até a conclusão desse trabalho. Contudo, a ferramenta foi executada completamente na plataforma *web*, ainda que em execuções locais, e a mesma se comportou da forma esperada, tanto no navegador Mozilla Firefox 33.1 quanto no Internet Explorer 11.0 e no Google Chrome 39.0.

O algoritmo de integração de dados entre os bancos de dados do NCBI e o banco de dados STRING também foi aprimorado. Na versão *web*, os identificadores das proteínas são consultados no banco de dados STRING apenas quando os mesmos não existem no banco de dados local da aplicação. Nas situações onde os identificadores das proteínas não se encontram no banco de dados local, mas são encontrados durante a execução do algoritmo aprimorado, eles estão sendo armazenados localmente, criando as referências diretas entre os dados dos bancos de dados do NCBI e do banco de dados STRING.

Analizando as redes PPI obtidas como resultado da execução da aplicação na versão *web*, a integração fez com que o resultado dessa versão da ferramenta não fosse exatamente igual ao da versão desktop. Entretanto, a versão *desktop* apresentava falhas para algumas proteínas, como a COX-1, onde a rede PPI para o organismo *Homo sapiens* não era retornada porque não encontrava seu identificador no banco STRING. Falha essa que a versão *web* da ferramenta não apresentou. Portanto, a alteração no algoritmo afetou a consulta da rede PPI, mas também melhorou o resultado obtido sendo possível encontrar identificadores de proteínas que na versão *desktop* não foi possível.

Uma melhoria a ser feita neste programa é permitir ao usuário armazenar o estado do *workflow* para ser recuperado posteriormente evitando que o usuário precise executar novamente as etapas do *workflow* que já foram executadas para um mesmo conjunto de dados. Na versão *web*, se ocorrer algum problema que venha a abortar a execução do *workflow*, o usuário é obrigado a executar todas as etapas novamente por não conseguir salvar e recuperar o estado da aplicação. Essa melhoria também facilitaria a realização de múltiplas análises sobre um conjunto de dados onde, a cada execução, poucas alterações nesses dados seriam efetuadas.

6. Referências

- Altschul, S.F. et al. (1990). Basic Local Alignment Search Tool – BLAST. *Journal Molecular Biology*, 215, 403-440.
- Altschul, S. F. et al. (1997). Gapped blast and Psi-Blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25, 3389–3402.
- Barnes, M. R. e Gray, I. C. *Bioinformatics for geneticists*. Wiley, 2003.
- Benson, D. A. et al. (2014). GenBank. In: *Nucleic Acids Research: Database issue*, Oxford, v. 42, n. 1, p.D32-D37, <http://nar.oxfordjournals.org/>, abril.
- Chirigati, F. e Freire, J. (2012). Towards Integrating Workflow and Database Provenance. Springer, *Provenance and Annotation of Data and Processes*, 7525, 11-23.
- Elmasri, R., Navathe, S. B. (2011). *Sistemas de banco de dados*. Pearson, 6^a edição.

- Gish, W. and States, D.J. 1993. Identification of protein coding regions by database similarity search. *Nature Genet.* 3:266-272.
- Jensen L. J., Kuhn M., Stark M., Chaffron S., Creevey C., Muller J., Doerks T., Julien P., Roth A., Simonovic M., Bork P., von Mering C. STRING 8 - A Global View on Proteins and their Functional Interactions in 630 Organisms. *PubMed*, 2009.
- Jones, N. C. and Pevzner, P. A. An introduction to bioinformatics algorithms. Massachusetts Institute of Technology. MIT Press books, 2004.
- Karsenti, E. (2012). Towards an ‘Oceans Systems Biology’. *Molecular Systems Biology*, 8, 575.
- Köhler, J., Philippi, S.; Lange, M. (2003). SEMEDA: ontology based semantic integration of biological databases. In: *Bioinformatics*, Oxford, v. 19, n. 18, p.2420-2427.
- Lesk, A. M. (2008). Introdução à Bioinformática. Artmed, 2^a edição.
- Lushbough, C. M., Gnimpyeba, E. e Dooley, R. (2013). BioExtract Server, a Web-based Workflow Enabling System, Leveraging iPlant Collaborative Resources. In: *Ieee International Conference On Cluster Computing (cluster)*, Los Alamitos, p.1-3.
- Mincato, D. O. (2014). Integração de Dados e Migração Web da Ferramenta Net2Homology. Monografia (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e da Tecnologia, Universidade de Caxias do Sul, Caxias do Sul.
- Notari, D. L. (2012). Desenvolvimento de workflows científicos para a geração e análise de diferentes redes de interatomas. Tese (Doutorado em Biotecnologia) – Programa de PósGraduação em Biotecnologia, Universidade de Caxias do Sul, Caxias do Sul.
- Puga, S., França, E., Goya, M. (2014). Banco de Dados: Implementação em SQL PL/SQL e Oracle 11g. Pearson, São Paulo.
- Silva, C., Anderson, E., Santos, E. and Freire, J. (2010). Using VisTrails and Provenance for Teaching Scientific Visualization. *Proceedings of the Eurographics Education Program*, 2010. Sommerville, I. Engenharia de Software. 8.ed. São Paulo: Pearson Addison-Wesley, 2007.
- Szklarczyk, D. et al. (2011). The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. In: *Nucleic Acids Research: Database issue*, Oxford, v. 39, n. 1, p.D561-D568.
- WfMC - Workflow Management Coalition, The Workflow Handbook 2004, Fischer,L.(ed.) Disponível em: <<http://www.wfmc.org/information/handbook04.htm>>.

Análise de Abordagens para Interoperabilidade entre Bancos de Dados Relacionais e Bancos de Dados NoSQL

Geomar A. Schreiner¹, Denio Duarte², Ronaldo dos Santos Mello¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

²Universidade Federal da Fronteira Sul (UFFS) – Campus Chapecó

schreiner.geomar@posgrad.ufsc.br, duarte@uffs.edu.br, r.mello@ufsc.br

Abstract. *Several applications produce and manipulate today a large volume of data, known as Big Data. Traditional databases, in particular, Relational Databases (RDBs), are not able to manage Big Data. Because of this, new data models have been proposed for manipulating large data sets, with focus on scalability and availability. Most of this data models belongs to the so-called NoSQL DBs. However, NoSQL DBs are not compatible, in general, with the SQL standard, and developers that use RDBs have to learn new models and access interfaces to develop Big Data-based applications. To deal with this problematic, approaches have been proposed to support the interoperability between SQL and NoSQL DBs. This work presents a comparative analysis of some related approaches for all people interested in this subject. We also intend to contribute to the research for a relational-to-NoSQL data and operations' interoperability solution.*

Resumo. *Diversas aplicações atualmente produzem e manipulam um grande volume de dados, denominados Big Data. Bancos de dados tradicionais, em particular, os Bancos de Dados Relacionais (BDRs), não são adequados ao gerenciamento de Big Data. Devido a isso, novos modelos de dados têm sido propostos para manipular grandes massas de dados, enfatizando a escabilidade e a disponibilidade. A maioria destes modelos de dados pertence a uma nova categoria de gerenciadores de dados denominados BDs NoSQL. Entretanto, BDs NoSQL não são compatíveis, em geral, com o padrão SQL e desenvolvedores que utilizam BDRs necessitam aprender novos modelos de dados e interfaces de acesso para produzirem aplicações baseadas em Big Data. Para lidar com esta problemática, abordagens têm sido propostas para o suporte da interoperabilidade entre BDRs e BDs NoSQL. Este artigo apresenta uma análise comparativa de algumas destas abordagens para todos aqueles interessados no assunto. Pretende-se ainda contribuir para a pesquisa direcionada ao desenvolvimento de uma solução para a interoperabilidade de dados e operações entre BDRs e BDs NoSQL.*

1. Introdução

Uma nova geração de aplicações surgiu para atender vários tipos de usuários, desde um pequeno grupo a grandes empresas. Essas aplicações acabaram por produzir e manipular um volume massivo de dados, dados esses denominados de *Big Data*. Exemplos são aplicações que gerenciam redes sociais e redes de sensores. Durante décadas, os

BDRs atenderam satisfatoriamente os requisitos de diversas aplicações, proporcionando simplicidade, robustez e desempenho na manipulação dos dados. Entretanto, estes BDs mostram-se ineficientes para a manipulação de Big Data, devido, por exemplo, ao comprometimento do seu desempenho com verificações de consistência de dados e com o processamento de consultas complexas que envolvem junções.

Com base nas carências dos BDRs foram propostos novos modelos de dados voltados a arquiteturas de sistemas de computação nas nuvens. Estas arquiteturas possuem como característica alta escalabilidade, baseada no crescimento horizontal dos sistemas, ou seja, a expansão da capacidade através da adição de novas infra-estruturas, como por exemplo, máquinas ou softwares. Ao contrário, os sistemas tradicionais que utilizam BDRs se baseiam em crescimento vertical da infra-estrutura, ou seja, para prover melhorias, o hardware e/ou software do sistema deve ser incrementado. Estes novos modelos de dados não tradicionais são denominados de BDs NoSQL (not only SQL) [Abadi 2009]. Exemplos destes novos modelos são os BDs *orientados a colunas* como CassandraDB e Cloudy, BDs *de documentos* como MongoDB e SimpleDB, e BDs *chave-valor* como o Voldemort.

BDs NoSQL não seguem o modelo relacional de representação dos dados e, portanto, o padrão de acesso SQL não é aplicável. Por outro lado, a maior parte das organizações utilizam BDRs para o gerenciamento dos seus dados e, caso elas desejem mudar para uma solução NoSQL devido ao aumento no volume dados a gerenciar, esta migração de modelo de dados e de interface de acesso é muito onerosa. Uma das principais justificativas deste alto custo é a baixa curva de aprendizagem que os BDs NoSQL proporcionam a seus usuários [Chung et al. 2013].

Desta forma, torna-se pertinente o desenvolvimento de soluções que auxiliem na redução deste custo. Duas abordagens são geralmente empregadas. Uma delas é o desenvolvimento de uma camada de software que seja capaz de receber instruções SQL e executá-las sobre um BD NoSQL [dos Santos Ferreira et al. 2013, Chung et al. 2013]. Outra abordagem é adaptar um sistema gerenciador de BDR (SGBDR) de forma que este manipule seus dados de maneira relacional, mas os armazene em um BD NoSQL [Egger 2009, Arnaut et al. 2011].

Neste contexto, este artigo apresenta uma análise de propostas existentes para interoperabilidade entre SGBDRs e SGBDs NoSQL, enfatizando o mapeamento de modelos e de algumas operações SQL para operações executáveis em SGBDs NoSQL. Este estudo leva em consideração apenas trabalhos que realizam o mapeamento no sentido Relacional para NoSQL, contribuindo para a pesquisa neste tema.

O restante deste artigo está organizado conforme segue. A Seção 2 descreve as abordagens relacionadas. A Seção 3 apresenta um comparativo entre elas e a Seção 4 é dedicada às conclusões.

2. Abordagens

Esta seção descreve as abordagens presentes na literatura. Elas são classificadas neste trabalho em duas categorias: *Layer* e *Storage Engine*. Abordagens do tipo *Layer* implementam uma camada de software que define uma abstração sobre um BDs NoSQL, permitindo ao usuário a definição e a manipulação dos dados utilizando instruções SQL.

Tabela Filmes			
id	nome	diretor	ano
1	Psyco	1	1960
11	The Godfather	2	1972
111	Patton	2	1970

Tabela Diretores		
id	nome	premios
1	Alfred Hitchcock	31
2	Francis Coppola	50

Figura 1. BD Relacional no Domínio de Cinema.

Já abordagens do tipo *Storage Engine* modificam o mecanismo de armazenamento de BDRs. Desta forma, os dados são criados e manipulados em um ambiente relacional, porém, armazenados em um BDs NoSQL. Para cada uma das abordagens são descritos o modelo de dados NoSQL utilizado, as regras de mapeamento propostas entre os modelos e as estratégias para processamento de junções. Todos os exemplos de mapeamento apresentados são baseados em um esquema relacional para um domínio de *Cinema* constituído pelas tabelas da Figura 1.

2.1. Abordagens do Tipo Layer

Abordagens classificadas como *Layer* traduzem comandos SQL para métodos de acesso específicos dos BDs NoSQL. Uma camada é implementada entre as requisições do usuário e a interface de acesso ao BD NoSQL. Ela é responsável por receber os comandos SQL, traduzí-los e executá-los sobre o BD NoSQL. Quatro propostas são descritas nesta Seção: *SimpleSQL*, que implementa uma camada sobre o SimpleDB [dos Santos Ferreira et al. 2013]; *JackHare*, que é uma camada sobre o HBase [Chung et al. 2013]; *Unity*, uma camada que permite acesso a dados armazenados em BDRs e BDs NoSQL na nuvem, mais especificamente o BD NoSQL MongoDB [Lawrence 2014]; e *Rith et al.*, que permite consulta a dados relacionais e NoSQL, em particular, os BDs MongoDB e Cassandra [Rith et al. 2014].

2.1.1. Estratégia de Mapeamento

SimpleSQL permite que usuários executem operações SQL DDL e DML sobre o SimpleDB. SimpleDB é um BD NoSQL com um modelo de dados é orientado a documento que possui os conceitos de *domínio*, *item*, *atributo* e *valor*. Um domínio é composto por um nome *dom* e um conjunto de itens *it_i* na forma $(dom, \{it_1, \dots, it_n\})$. Cada item é composto de um nome *name* e uma coleção *m* de atributos na forma $\{name: \{key_1 : value_1; \dots; key_m : value_m\}\}$, sendo cada *key_i* o nome de um atributo (sua chave de acesso) e *value_i* o valor de *key_i*. O SimpleSQL mapeia um BDR *db* em um domínio com nome *db*. Cada um dos itens deste domínio representa um chave primária *pk* (o nome do item, i.e., *name*) de uma tabela *t* de *db*, ou seja, cada linha de *t* é representada por um item do SimpleDB. Para cada item o SimpleSQL cria um atributo especial chamado *SimpleSQL_TableName* que armazena o nome de *t*. Por exemplo, a tabela *Filmes* seria armazenada no domínio *Cinema* como: $\{1 : \{SimpleSql_TableName : "filmes"; nome : "Psycho"; diretor : "1"; ano : "1960"\}\}, \{11 : \{SimpleSQL_TableName : "filmes"; nome : "$



Figura 2. Tabelas do BD Cinema mapeadas para o HBase através do JackHare.

“*TheGodfather*”; diretor : “2”; ano : “1972”} } e {111 : {SimpleSQLTableName : “filmes”; nome : “Patton”; diretor : “2”; ano : “1970”}}.

JackHare é um *framework* composto por um compilador SQL, um driver JDBC e um método que utiliza a tecnologia *Map-Reduce* [Dean and Ghemawat 2008] para acessar dados armazenados no HBase. HBase é um BD NoSQL que segue o modelo de dados colunar. Um BD neste modelo é composto por um *namespace*. Cada *namespace* possui uma série de *HTables* que, por sua vez, possuem um conjunto de *famílias de colunas*. Cada família de coluna possui um conjunto de chaves compostas por colunas e valores. Não é requerido que todas as ocorrências de uma HTable tenham as mesmas colunas.

O mapeamento realizado pelo JackHare segue as seguintes regras: (i) cada BDR db é mapeado para uma HTable K , com o mesmo nome de db ; (ii) cada tabela t pertencente a db é mapeada para uma família de colunas f com o mesmo nome de t ; (iii) cada linha r de t é mapeada para um conjunto de pares chave-valor K que representam as suas colunas e respectivos valores. A chave deste conjunto é obtida concatenando a chave primária de r com o nome de t ; (iv) uma coluna c de r é mapeada para uma coluna C pertencente à K .

A Figura 2 ilustra o resultado do mapeamento do BD Cinema para o HBase. As bordas mais externas representam as famílias de colunas *Filmes* e *Diretores*. As tuplas da tabela *Filmes*, por exemplo, estão representadas na família de colunas *Filmes*. A borda hifenizada indica as colunas e seus valores. As chaves primárias foram separadas na figura para enfatizar como a abordagem as constrói.

Outra abordagem que implementa uma camada relacional é o *Unity*. Esta abordagem utiliza o BD NoSQL MongoDB para manter os seus dados. MongoDB também é um BD orientado a documentos, sendo seu modelo de dados composto pelos conceitos de *base de dados*, *coleções de documentos* e *documentos*. Documentos são compostos por um conjunto de pares chave-valor. Unity realiza o mapeamento de um BDR db para um BD MongoDB M_{db} com o mesmo nome de db . Cada uma das coleções M_c de M_{db} representa uma tabela t de db . Cada linha r de t é um documento M_d em M_c . Atributos e valores de r são representados por pares $nome_atributo : valor_atributo$. Por exemplo, a tabela *Filmes* é convertida para a seguinte coleção de documentos: {1 : {nome : “Psycho”; diretor : “1”; ano : “1960”}}, {11 : {nome : “TheGodfather”; diretor : “2”; ano : “1972”}} e {111 : {nome : “Patton”; diretor : “2”; ano : “1970”}}.

Rith et al. apresenta uma abordagem que permite a execução de consultas SQL no MongoDB e no Cassandra. Para tanto, são criados conectores que realizam o *parsing* de consultas SQL, enviam para o BD alvo e retornam o resultado. Não são fornecidos detalhes do mapeamento de instruções de consulta SQL na literatura a respeito de *Rith et al.*

al.. Mesmo assim, imagina-se que este mapeamento seja trivial para consultas envolvendo apenas uma tabela, uma vez que ambos os BDs alvo possuem uma linguagem de consulta semelhante em sintaxe ao padrão SQL.

2.1.2. Processamento de Junções

SGBDs NoSQL têm suas operações geralmente baseadas em registros individuais, diferentemente dos SGBDs Relacionais onde as operações são baseadas em conjuntos, incluindo operações de junção. Assim sendo, enquanto o padrão SQL permite consultas complexas com junções, SGBDs NoSQL não suportam tal funcionalidade, cabendo à solução de interoperabilidade tratar o processamento de junções.

No caso do *SimpleSQL*, uma consulta é decomposta em uma lista de atributos, tabelas, junções e predicados. Caso existam junções, ele os divide em consultas simples com predicados, se necessário. Após obter o resultado de cada uma destas consultas, é criada uma tabela que respeita o esquema do resultado esperado. Os itens de resposta do resultado são combinados por similaridade utilizando as chaves primárias e estrangeiras do esquema relacional, tomando por base as informações de mapeamento entre os esquemas relacional e de documento do SimpleDB. O desempenho do processamento das junções é, obviamente, dependente do volume de dados envolvido.

JackHare permite que usuários executem consultas envolvendo junções utilizando-se de operações Map-Reduce. Inicialmente, é realizada uma verificação do tamanho das tabelas. A menor das tabelas é transformada em uma lista de chaves na fase de Map. A lista de chaves é armazenada no HBase. Na sequência, combinam-se os valores da tabela maior com a lista de chaves, atualizando a lista. Este processo também é feito com operações Map-Reduce a fim de se valer do paralelismo para obter ganho de desempenho. Após todas as entradas terem sido combinadas, a abordagem retorna a lista de chaves já combinadas, que constitui a fase de Reduce. Esta técnica permite que o JackHare manipule grandes volumes de dados.

Unity possibilita o acesso integrado a diversas fontes de dados na nuvem, sejam relacionais ou BDs MongoDB. Assim sendo, se a junção for executada sobre uma única fonte relacional, a junção é processada normalmente. Caso a fonte seja um BD NoSQL, a abordagem implementa um *hash join*, com base nas chaves dos registros, para executar a operação. Caso os dados alvo estejam armazenados em duas fontes, o processador toma uma decisão baseado no tamanho das coleções de dados. Se ambas forem grandes, os dados são extraídos em paralelo e o *Unity* executa também um *hash join*. Se os dados de uma fonte forem substancialmente maiores que a outra, ele extrai os dados da menor e os utiliza como filtro na outra fonte, ou ainda copia os dados para uma tabela temporária em um BDR e então executa a junção.

A abordagem proposta por *Rith et al.* não suporta consultas com junções até o momento da escrita deste artigo.

2.2. Abordagens do Tipo Storage Engine

Abordagens nesta categoria modificam a forma pela qual o SGBD armazena seus dados, permitindo o armazenamento em um formato NoSQL. Neste caso, a camada física

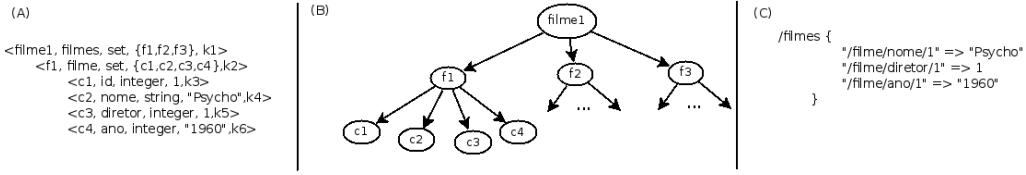


Figura 3. Mapeamento da Tabela Filmes para o Phoenix.

do SGBD relacional é alterada. Abordagens nesta categoria devem ser capazes de realizar quatro operações sobre a camada de armazenamento NoSQL: criar, ler, atualizar e deletar dados. Assim sendo, o SGBD realiza requisições para o mecanismo de armazenamento modificado, este mecanismo mapeia os dados relacionais para o modelo NoSQL e executa a operação. Nesta categoria foram encontradas 3 abordagens: *Phoenix* [Arnaud et al. 2011], *CloudyStore* [Egger 2009] e *DQE* [Vilaça et al. 2013].

2.2.1. Estratégia de Mapeamento

Phoenix implementa uma nova *storage engine* para o SGBD MySQL armazenando seus dados no *Scalaris*. *Scalaris* é um banco NoSQL baseado no modelo chave-valor. Neste modelo, a chave é um identificador único para um elemento (item de dado) e o valor é uma descrição do elemento. O valor pode ser de dois tipos: simples ou complexo. Um valor simples é um valor atômico (*e.g.*, string e integer). Já um valor complexo é constituído por um conjunto de pares chave-valor.

Phoenix define um modelo de dados intermediário para realizar o mapeamento relacional-chave-valor, chamado VOEM (*Value-based OEM*). VOEM é uma extensão do modelo OEM (*Object Exchange Model*) [Papakonstantinou et al. 1995]. OEM é um modelo de descrição de objetos, sendo que cada objeto instanciado possui um identificador único (*oid*). VOEM estende as capacidades do OEM com a noção de chave, ou seja, um subconjunto de atributos do objeto que identificam um objeto através de valores. Desta forma, é possível identificar uma tupla.

Um objeto VOEM é descrito pela sétupla $v = \langle oid, \lambda, \tau, \nu, k \rangle$, onde o *oid* é o identificador do objeto, λ é o rótulo do objeto, τ indica o tipo do objeto (simples ou complexo), e ν é o valor do objeto. Se o tipo do objeto for complexo, então ν contém um conjunto de *oid*. A última parte de um objeto VOEM k é uma chave que identifica unicamente um par chave-valor. A Figura 3(A) apresenta a codificação VOEM da tabela *Filmes*.

VOEM permite o mapeamento do modelo relacional para o modelo chave-valor. Este mapeamento é baseado nas seguintes regras: (i) uma linha r de uma tabela relacional t é representado por um objeto VOEM O_i com rótulo t e seu valor é uma coleção de *oids* que mapeiam todas as colunas de r ; (ii) cada coluna c de tipo τ de r é mapeada para para um objeto VOEM O_i com rótulo c e tipo τ .

A coleção de objetos é transformada em um grafo VOEM rotulado V (Figura 3(B)). Cada vértice de V é um objeto e cada uma das arestas um relacionamento entre objetos. Este grafo é usado para criar a chave de cada objeto VOEM O . Para definir a chave, é feita uma busca em profundidade cuja origem é o primeiro nodo de V e destino



Figura 4. BD Cinema mapeado pelo CloudyStore.

o vértice O . Na Figura 3(C), o par chave-valor ”/filme/nome/1 => Psycho”, tem sua chave definida através da concatenação dos rótulos do caminho do objetos VOEM $f1$ (*filme*), $c2$ (*nome*) (Figura 3(B)) e do valor 1 da chave primária (representada no grafo como $c1$) que identifica a linha mapeada.

Por fim, o mapeamento de VOEM para o modelo chave-valor é mais intuitivo: um objeto VOEM O é mapeado para um par chave-valor kv , onde a chave é a chave de O . Se o valor de O for simples, então, o valor de kv será o mesmo de O . Se o valor for complexo, o valor de kv será o conjunto de valores dos objetos VOEM que formam o valor de O mapeados para o modelo chave-valor. Por exemplo, o objeto “f1”(Figura 3(A)) é apresentado já mapeado no modelo chave-valor na Figura 3(C).

CloudyStore é uma abordagem similar à *Phoenix*, porém realiza o mapeamento para um BD NoSQL colunar denominado *Cloudy* [Egger 2009]. O modelo de dados do *Cloudy* é similar ao do Hbase, ou seja, composto por *keyspaces*, família de colunas, chaves, colunas e valores.

CloudyStore propõe três regras para mapear um BDR para o modelo do *Cloudy*: (i) um BD db é mapeado para uma *keyspace* K com o nome db ; (ii) uma tabela t de db é mapeada para uma família de colunas com o mesmo nome de t ; (iii) uma linha r de t é mapeada para um conjunto de pares chave-valor rk de K , sendo a chave de rk uma chave interna do MySQL para r (conhecida como *rowid*). Cada coluna c de r é mapeada para uma coluna ck pertencente à rk , cujo nome será c .

A Figura 4 apresenta o mapeamento do BD *Cinema*. As tabelas *Filmes* e *Diretores* são mapeadas para as famílias de colunas *Filmes* e *Diretores*, respectivamente. Cada linha de uma tabela é mapeada para pares *chave-valor*. Por exemplo, a primeira linha de *Filmes* é mapeada para um valor `{"nome" : "Psyco", "diretor" : "1", "ano" : "1960"}` na família de coluna *Filmes* (borda com hifens na Figura 4). As chaves a , b , c , d e e para as colunas das famílias (borda pontilhada na Figura 4) representam os *rowids* que identificam as tuplas no MySQL.

DQE é uma abordagem que define uma storage engine para o SGBDR Derby, armazenando os dados no BD colunar HBase. Seu mapeamento é baseado em 4 regras: (i) um BD db é mapeado para uma *keyspace* K com o nome de db ; (ii) cada tabela t de db é mapeada para uma HTable ht nomeada com t ; (iii) uma linha r de t é mapeada para um conjunto de pares chave valor rk de ht ; (iv) cada coluna c de r é mapeada para uma coluna ck pertencente à rk , com o nome de c . Se o BD possuir atributos indexados, *DQE* cria, para cada atributo, uma nova HTable, sendo os valores do atributo mapeados

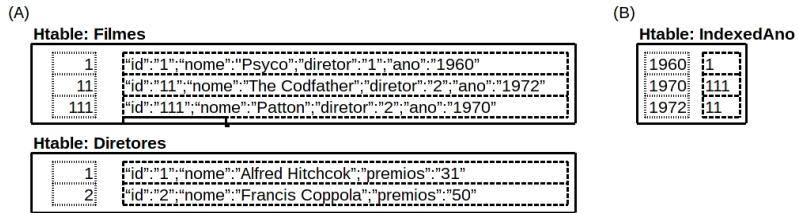


Figura 5. BD Cinema mapeado pela Abordagem DQE.

para uma linha nesta tabela. A chave da linha é uma coluna que mantém o valor indexado e uma outra coluna atua como identificador da rk , indicando a linha da tabela a qual o valor indexado pertence. A Figura 5 apresenta o mapeamento para o BD *Cinema*. O mapeamento gera 3 HTables: as duas primeiras (Figura 5(A)) correspondem às tabelas *Filmes* e *Diretores*, enquanto a terceira (Figura 5(B)) representa o mapeamento de um índice criado para o atributo *ano* da tabela *Filmes*. Nesta tabela de índice, a chave da linha é o atributo indexado (*ano*) e o valor é a chave da coluna que possui este valor de ano na respectiva HTable. Se o atributo indexado não possuir um valor único, então uma HTable diferente é criada para o atributo indexado, onde o atributo indexado será a chave e para cada ocorrência do atributo indexado é criado um par coluna e valor, onde a coluna é a chave do valor na respectiva HTable e o valor é vazio.

2.2.2. Processamento de Junções

As abordagens do tipo *Storage Engines* não implementam estratégias para o processamento de junções, utilizando as técnicas implementadas pelos SGBDRs. O SGBDR cria o plano de consulta e o otimiza, verifica as junções a serem executadas e somente faz requisições às *storage engines* para buscar os dados envolvidos na consulta. Desta forma, o SGBDR é o responsável por este processamento.

3. Análise Comparativa

A Tabela 1 apresenta um comparativo das abordagens apresentadas neste artigo para a interoperabilidade entre BDRs, baseados no padrão de acesso SQL, e BDs NoSQL com seus diversos modelos de dados não-relacionais, que não adotam em geral este padrão. As seguintes características são consideradas na comparação: (i) o tipo da abordagem (*Layer ou Storage Engine*); (ii) o modelo de dados NoSQL que a abordagem interopera; (iii) quais instruções SQL a abordagem suporta (as abordagens JackHare e Rith *et al.* não implementam todos as instruções DML por isso são marcada como DML Restrita); (iv) se a abordagem apresenta suporte para o processamento de requisições em mais de uma fonte de dados; (v) se a abordagem utiliza um dicionário de dados para manter informações de mapeamento; e (vi) como a abordagem processa junções, se for o caso.

A escolha da abordagem a ser utilizada pode depender fortemente da sua categoria. Cada uma delas apresenta vantagens e desvantagens. Abordagens baseadas em *layers* podem, em alguns casos, acessar mais de uma fonte de dados e tendem a ser mais flexíveis pois são mais facilmente estendidas para o acesso a outros SGBDs NoSQL, uma vez que essa extensão requer alteração em uma camada de software independente dos SGBDs envolvidos (relacional e NoSQL). Por outro lado, elas podem ter problemas de

Abordagem	Categoria	Modelo NoSQL	Suporte SQL	Múltiplas Fontes	Dicionário	Junções
SimpleSQL	Layer	Documento	DDL+DML	Não	Sim	Similaridade
JackHare	Layer	Colunar	DML Restrita + DDL	Não	Sim	Map-Reduce
Unity	Layer	Documento	DML	Sim	Sim	Hash-Join
Rith et al.	Layer	Colunar/Documento	DML Restrita + DDL	Sim	-	-
Phoenix	Storage Engine	Chave-valor	Instruções MySQL	Não	Não	SGBD
CloudyStore	Storage Engine	Colunar	Instruções MySQL	Não	Sim	SGBD
DQE	Storage Engine	Colunar	DDL+DML	Não	Sim	SGBD

Tabela 1. Comparativo das Abordagens Analisadas.

desempenho no processamento de grandes volumes de dados, em particular, na execução de consultas complexas. Já abordagens baseadas em *storage engines* tendem a ser mais escaláveis, uma vez que são implementadas no *kernel* do SGBDR. Entretanto, possuem a desvantagem de estarem atreladas a um SGBDR, perdendo em flexibilidade.

Com relação ao modelo de dados, verifica-se a interoperabilidade para diversos modelos NoSQL. Entretanto, a grande maioria das abordagens é limitada ao mapeamento para apenas um modelo, inexistindo ainda uma solução genérica para o mapeamento SQL-NoSQL. Ainda com relação ao mapeamento, percebe-se que as abordagens baseadas em *storage engines*, por estarem implantadas em um SGBDR específico, suportam, em geral, o mapeamento de qualquer operação SQL disponibilizada por este SGBDR. Já as abordagens baseadas em *layers* limitam-se a mapear apenas algumas instruções SQL DML e/ou DDL, devido ao custo de implementar a transformação de cada uma delas para o SGBD NoSQL alvo.

A presença de um dicionário é um requisito fundamental para abordagens baseadas em *layers*, pois a gerência do mapeamento é feito por uma camada de software específica para a interoperabilidade e tal informação é vital para proceder as conversões estruturais e de instruções. Mesmo assim, verifica-se que boa parte das abordagens da outra categoria também o consideram.

Por fim, um desafio na efetivação da interoperabilidade entre o modelo relacional e modelos NoSQL é o suporte a junções, inexistente em BDs NoSQL para não comprometer a disponibilidade e a escalabilidade, e que deve ser provida pela abordagem. Conforme indicado na Tabela 1, abordagens da categoria *storage engine* utilizam o SGBDR para isso. Já as abordagens baseadas em *layer* utilizam diversas técnicas, não existindo um consenso.

4. Conclusão

BDRs tornaram-se inadequados à aplicações que lidam com *Big Data*. Para atender a essa demanda crescente, novas soluções de gerenciamento de dados, os denominados BDs NoSQL, estão surgindo. Apesar das vantagens trazidas por tais soluções, muitos destes SGBDs não suportam o acesso SQL a dados. Este é um impedimento para um grande número de aplicações que utilizam SGBDRs e não desejam arcar com um alto custo de migração e curva de aprendizado para tecnologias NoSQL.

Assim sendo, soluções para a interoperabilidade relacional-NoSQL são relevantes e este artigo contribui para esta temática através de uma análise comparativa de abordagens relacionadas que visa ser um referencial de informação e de escolha de abordagem mais adequada para determinada aplicação.

Através desta análise percebe-se alguns pontos em aberto nesta temática, como

a ausência de uma abordagem que realize um mapeamento completo do padrão SQL para um determinado modelo de dados NoSQL, bem como uma abordagem genérica, que permita a interoperabilidade para qualquer modelo de dados NoSQL. Na linha desta pesquisa pretende-se ainda, como trabalho futuro, realizar uma avaliação comparativa de desempenho das mesmas.

Referências

- Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. *IEEE Data Eng. Bull.*, 32(1):3–12.
- Arnaut, D. E., Schroeder, R., and Hara, C. S. (2011). Phoenix: A relational storage component for the cloud. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 684–691. IEEE.
- Chung, W.-C., Lin, H.-P., Chen, S.-C., Jiang, M.-F., and Chung, Y.-C. (2013). Jackhare: a framework for SQL to NoSQL translation using MapReduce. *Automated Software Engineering*, pages 1–20.
- Dean, J. and Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113.
- dos Santos Ferreira, G., Calil, A., and dos Santos Mello, R. (2013). On providing DDL support for a relational layer over a document NoSQL database. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services, IIWAS ’13*, pages 125:125–125:132, New York, NY, USA. ACM.
- Egger, D. (2009). *SQL in the Cloud*. PhD thesis, Master Thesis ETH Zurich, 2009.
- Lawrence, R. (2014). Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB. In *Computational Science and Computational Intelligence (CSCI), 2014 International Conference on*, volume 1, pages 285–290.
- Papakonstantinou, Y., Garcia-Molina, H., and Widom, J. (1995). Object exchange across heterogeneous information sources. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 251–260.
- Rith, J., Lehmayr, P. S., and Meyer-Wegener, K. (2014). Speaking in tongues: SQL access to NoSQL systems. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 855–857. ACM.
- Vilaça, R., Cruz, F., Pereira, J., and Oliveira, R. (2013). An effective scalable SQL engine for NoSQL databases. In *Distributed Applications and Interoperable Systems*, pages 155–168. Springer.

Modelando Banco de Dados Relacionais e Geográficos Utilizando a Ferramenta GenDBM Tool

João Victor Guinelli¹, André de Souza Rosa¹, Carlos Eduardo Pantoja²

¹CEFET/RJ - UnED Nova Friburgo

Av. Gov. Roberto da Silveira, 1900 - Prado - 22.635-000 - Nova Friburgo - RJ - Brasil

²CEFET/RJ - UnED Maria da Graça

Rua Miguel Ângelo, 96 - Maria da Graça - 20.785-220 - Rio de Janeiro - RJ - Brasil

jvguinelli@gmail.com, andre_souza.rosa@hotmail.com, pantoja@cefet-rj.br

Abstract. This paper presents a tool for database modeling constructed based on MDA approach called GenDBM Tool, that is a set of plug-ins for the Eclipse IDE based on the conceptual modeling of relational and geographic database, thus the code generation following the SQL/SFS and ANSI SQL standards. For this, it is used a generic metamodel, which gathers concepts from the geographic and relational modeling notation, that are defined according with the metamodel and submitted to a set of M2T transformations rules, generating the code for the database implementation. This paper also presents some examples using the GenDBM Tool and compares its principal features with other tools that have the same purpose.

Resumo. Este artigo apresenta uma ferramenta para modelagem de banco de dados construída com base na abordagem MDA chamada GenDBM Tool, que é um conjunto de plug-ins para o IDE Eclipse, voltada para modelagem conceitual de banco de dados relacional e geográfico, assim como a geração de código no padrão ANSI SQL/DDL e SQL/SFS. Para isso é utilizado o GEDBM, meta-modelo genérico que reúne conceitos das notações de modelagem para banco de dados relacionais e geográficos, sendo os modelos definidos de acordo com o mesmo e submetidos a um conjunto de regras de transformação M2T, gerando o código para a implementação da base de dados. São apresentados alguns exemplos de utilização da GenDBM Tool e traçados comparativos das suas principais características em relação a algumas ferramentas voltadas para o mesmo fim.

1. Introdução

O projeto de banco de dados consiste na criação de modelos em diferentes níveis de abstração, e a partir de transformações entre esses modelos, chegar à implementação efetiva do banco de dados em um Sistema Gerenciador de Banco de Dados (SGBD) específico, utilizando uma linguagem de consultas, e.g. SQL [Elmasri et al., 2005]. A *Model-Driven Architecture* (MDA) é uma abordagem para construção de *software* através do processo de transformação entre modelos em diferentes níveis de abstração. O processo parte de modelos com maior nível de abstração até modelos que tratem das características específicas de implementação do *software* [OMG, 2003].

Existem diversas ferramentas para modelagem de banco de dados como a EERCASE [Fidalgo et al., 2013] e OMT-G Designer [Lizardo and Davis, 2014].

Contudo tais ferramentas são específicas para determinadas notações de modelagem conceitual, limitando o projetista na escolha da linguagem de modelagem. Tais ferramentas também não possuem um meta-modelo unificado, dificultando a extensão de outras notações às suas soluções. A geração da codificação é realizada direto do modelo específico para o código, dessa forma a integração de novos modelos ou notações implicaria em um novo conjunto de regras de geração.

Esse trabalho apresenta as funcionalidades e exemplos da ferramenta GenDBM Tool (disponível em <https://sourceforge.net/projects/gendbmtool/>) e de ferramentas relacionadas com o propósito de auxílio na modelagem de banco de dados relacionais e geográficos. A GenDBM Tool utiliza a MDA para prover um ambiente de modelagem de banco de dados relacionais e geográficos expansível a diferentes linguagens de modelagens e notações, além de permitir a geração de codificação para os padrões ANSI SQL e SQL *Simple Features Specification* (SFS) independente da escolha do projetista na fase de modelagem conceitual. O SFS é uma especificação que estende o padrão SQL permitindo o armazenamento, consulta e recuperação de dados com características espaciais, como formas geométricas e coordenadas geográficas [Borges et al., 2001].

A GenDBM Tool utiliza também o *Generic Database Metamodel* (GEDBM) [Rosa et al., 2013] para modelagem de banco de dados que reúne os conceitos das principais linguagens de modelagens relacionais e de uma notação de modelo geográfico, permitindo a criação de ferramentas gráficas e a integração de ferramentas já existentes em um único meta-modelo.

A ferramenta possui uma interface gráfica para a definição de modelos segundo o Modelo Entidade-Relacionamento. Para modelagens gráficas destinadas a Banco de Dados Geográficos (BDG), é possível utilizar a ferramenta OMT-G Design [Martinez and Frozza, 2014] para definição de modelos através da notação OMT-G e utilizar o modelo instanciado como entrada na GenDBM Tool para executar a geração de código.

Este trabalho está estruturado da seguinte forma: na seção 2 são fornecidos mais detalhes sobre a ferramenta e suas principais características; na seção 3 a utilização da ferramenta é exposta através de um exemplo de modelagem conceitual relacional e de uma modelagem conceitual geográfica; na seção 4 são discutidos alguns trabalhos relacionados; e finalmente na seção 5 apresenta-se a conclusão e alguns trabalhos futuros.

2. A Ferramenta

A ferramenta MDA exposta nesse trabalho foi elaborada com base em duas metodologias, uma proposta por [Rosa et al., 2013] e outra por [Guinelli et al., 2014], sendo adotada inicialmente a proposta por [Rosa et al., 2013], que define um meta-modelo genérico, o GEDBM, reunindo conceitos de modelagem conceitual de distintas notações de modelagem para Bancos de Dados Relacionais (BDR) como Modelo Entidade-Relacionamento (MER), *Crow'sFoot* e IDEF1X. Essa metodologia também define regras de transformação de modelo para texto, possibilitando que modelos instanciados a partir do GEDBM possam ser transformados em código de texto no padrão ANSI SQL/DDL 92/99/03.

A abordagem MDA para desenvolvimento de *softwares* tem por características principais o agrupamento de modelos por nível de abstração e os processos de

transformação aos quais esses modelos são submetidos para a efetiva implementação do sistema. Os níveis de abstração são divididos entre três grupos: os de maior nível de abstração, chamados Modelos Independentes de Computação (*Computer Independent Model* - CIM), que retratam as especificações do sistema sem levar em conta detalhes computacionais, os Modelos Independentes de Plataforma (*Platform Independent Model* - PIM), que descrevem a estrutura do sistema e sua parte lógica sem levar em conta detalhes pertinentes ao ambiente ou tecnologia utilizada para implementação do mesmo e o Modelo de Plataforma Específica (*Platform Specific Model* - PSM), que contém as especificações da plataforma onde o sistema será implementado [OMG, 2003].

O processo de transformação a qual os modelos envolvidos são submetidos no desenvolvimento do sistema pode ser manual ou automático, partindo do maior nível utilizado de abstração até a implementação do sistema. As transformações podem ocorrer de Modelo para Modelo (*Model To Model* - M2M) ou de Modelo para Texto (*Model To Text* - M2T), e são definidas a partir de especificações ou linguagens voltadas para esse fim [Mellor et al., 2005].

O conjunto de modelos de maior nível de abstração utilizado na metodologia adotada pela ferramenta [Rosa et al., 2013] é o PIM, que corresponde às notações de modelagem conceitual que possuem seus conceitos mapeados no GEDBM. Após a definição de um modelo, é possível aplicar regras de transformação de modelo para texto, transformando o modelo definido em um artefato de texto que contenha os detalhes da tecnologia utilizada para implementação da base de dados, sendo esse artefato correspondente ao PSM. As regras de transformação responsáveis por essa geração de código a partir do modelo conceitual se relacionam de acordo com o mostrado abaixo, na Figura 1.

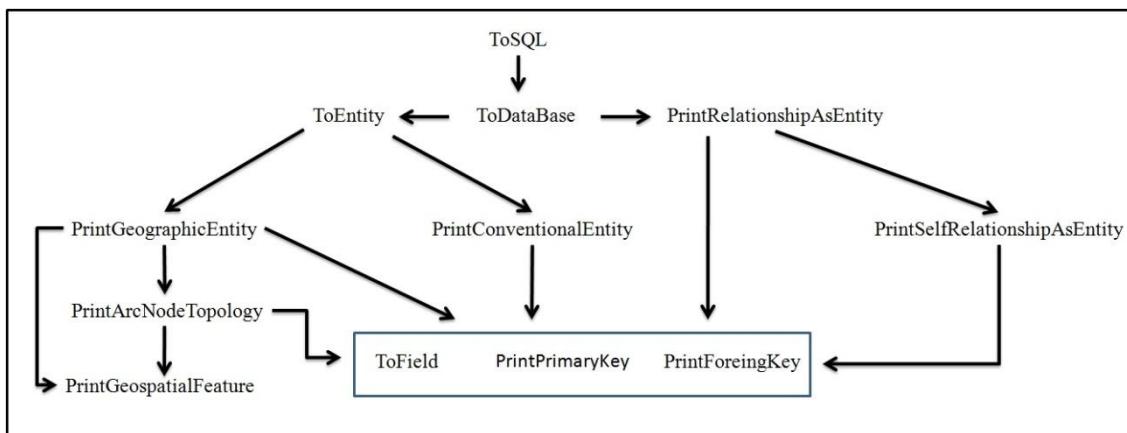


Figura 1 – Relacionamento entre as regras de transformação M2M.

As regras de transformação apresentadas acima foram especificadas utilizando o Acceleo [Obeo, 2012], cada uma dessas regras corresponde a um *template* nesta linguagem e possuem responsabilidades bem definidas. O *template ToSQL* recebe o meta-modelo instanciado e gera o arquivo *.txt* que irá receber o código SQL gerado pelas regras de transformação seguintes. Depois de criado este arquivo, o *template ToDataBase* é chamado. Este *template* passa todas as entidades instanciadas no meta-modelo para o *ToEntity* e os relacionamentos que resultam em entidades, como o relacionamento muitos para muitos, para o *PrintRelationshipAsEntity*.

O template *PrintRelationshipAsEntity* realiza ainda uma análise a fim de verificar se o relacionamento recebido é um auto-relacionamento, pois, caso seja, o código SQL será gerado pelo template *PrintSelfRelationshipAsEntity*. Além disso, ambos utilizam os templates *ToField*, *PrintPrimaryKey* e *PrintForeingKey* para gerar o código de seus atributos, chaves primárias e chaves estrangeiras.

Já o template *ToEntity* é responsável por analisar todas as entidades e definir o momento correto de criação das mesmas, de forma a garantir que não sejam criadas entidades contendo chaves estrangeiras que apontem para o campo de alguma outra entidade ainda não criada. Ele também é responsável por analisar cada uma destas entidades a fim de saber se as mesmas são do tipo geográfico: caso sejam, elas são passadas para o template *PrintGeographicEntity*; caso não sejam, elas são passadas para o template *PrintConventionalEntity*.

Estes dois últimos templates citados possuem basicamente as mesmas responsabilidades, a única diferença é que o ao contrário do segundo, que gera o código SQL para entidades convencionais, o primeiro gera o código SQL para entidades do tipo geográfico. Ambos passam a responsabilidade pela geração do código SQL referente aos atributos, chaves primárias e chaves estrangeiras para os templates *ToField*, *PrintPrimaryKey* e *PrintForeingKey*, respectivamente.

Adicionalmente, o template *PrintGeographicEntity* ainda solicita ao *PrintGeographicFeature* que gere o código do atributo geográfico que melhor represente o tipo geográfico da entidade analisada, ou, caso esta entidade represente uma topologia arco-nó a responsabilidade por gerar seu código SQL será do template *PrintArcNodeTopology*.

O GEDBM permite que não haja atrelamento da ferramenta implementada a uma única notação de modelagem conceitual para BDR por possuir em sua definição conceitos comuns presentes nas principais notações (*Crow'sFoot*, MER e IDEF1X), tornando-o aderente a essas modelagens. Essa característica confere liberdade de escolha ao projetista de banco de dados e facilita o trabalho em conjunto quando, em uma equipe de desenvolvimento, indivíduos dominam diferentes tipos de notação de modelagem. Além disso, o código gerado a partir das regras de transformação segue o padrão ANSI SQL, o que facilita sua utilização em diferentes Sistemas Gerenciadores de Banco de Dados (SGBD). Apesar de cada SGBD aplicar alguma característica distinta na sintaxe do código SQL que utiliza, bastaria apenas uma alteração na regra de transformação que trate da característica em questão ou no código gerado, para que o mesmo se torne aderente ao formato adotado pelo SGBD em questão.

Apesar das vantagens apresentadas, a metodologia proposta por [Rosa et al., 2013] não contempla modelagens voltadas para banco de dados geográfico. Foi proposta então, por [Guinelli et al., 2014], a extensão do GEDBM para torná-lo aderente a modelagens de componentes espaciais, definidas utilizando a notação OMT-G. Também foram definidas novas regras de transformação tornando a geração de código aderente à especificação SFS para o SQL/DDL [Guinelli et al., 2014].

A metodologia da GenDBM Tool também prevê mapeamento entre modelos através de definição de regras transformação de modelo para modelo, mapeando os conceitos presentes entre o meta-modelo da ferramenta OMT-G Design e o GEDBM. Caso um meta-modelo utilizado por outra ferramenta seja aderente ao da GEDBM, é possível definir regras de transformação que converta este modelo no utilizado pelo

núcleo da ferramenta, sendo assim possível dar prosseguimento no processo de geração de código.

A GenDBM Tool possui interface gráfica que permite a definição de modelos para BDR segundo a notação MER, também sendo possível utilizar a ferramenta OMT-G Design para realizar a definição de uma modelagem para Banco de Dados Geográficos (BDG) devido ao mapeamento entre modelos. A vantagem em gerar o artefato de texto através da GenDBM Tool mesmo havendo a possibilidade de fazê-lo através da OMT-G Design é que o código gerado através dela é direcionado ao PostgreSQL [Martinez and Frozza, 2010] enquanto o gerado pela GenDBM Tool, ao ser mais próximo dos padrões ANSI SQL e SFS/SQL se torna de mais fácil aplicação em outros SGBD.

3. Exemplos

Nessa seção, são apresentados: um exemplo de modelagem conceitual para BDR através da interface gráfica da ferramenta, utilizando a notação MER, e um exemplo de modelagem para BDG utilizando a notação OMT-G através da interface gráfica da ferramenta OMT-G Design. No caso do modelo segundo a notação OMT-G é necessário realizar o mapeamento entre o modelo definido e o GEDBM para a efetiva geração do código.

3.1. Modelagem para Bancos de dados Relacionais

A Figura 2 mostra a interface gráfica utilizada para BDR da ferramenta. Na barra lateral direita se localiza a paleta de ferramentas com os componentes disponibilizados para realizar a definição de modelos segundo a notação MER. À esquerda têm-se os componentes utilizados para o funcionamento da ferramenta e que guardam as configurações e dados dos modelos definidos a partir da mesma.

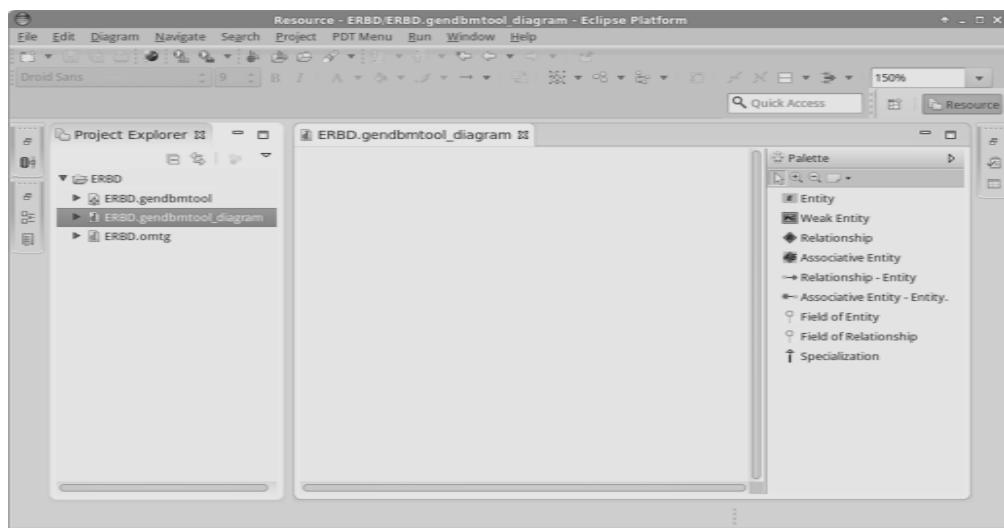


Figura 2 – Interface gráfica da GenDBM Tool.

Para exemplificar a modelagem conceitual para BDR, é tomado como referência um modelo definido por [Chen, 1976]. O domínio representado se trata de uma fábrica, onde se deseja armazenar os dados sobre os funcionários (*Employee*), sobre seus dependentes (*Dependent*), sobre os projetos (*Project*), sobre as partes utilizadas pelos projetos (*Part* e *Supplier*) e como essas entidades interagem entre si.

O modelo definido através da ferramenta pode ser visto na Figura 3. Para representar os relacionamentos de muitos para muitos foi utilizada a estrutura de entidade associativa, oferecida pela ferramenta, com o intuito de explicitar essa funcionalidade tanto durante a modelagem conceitual quanto na geração de código.

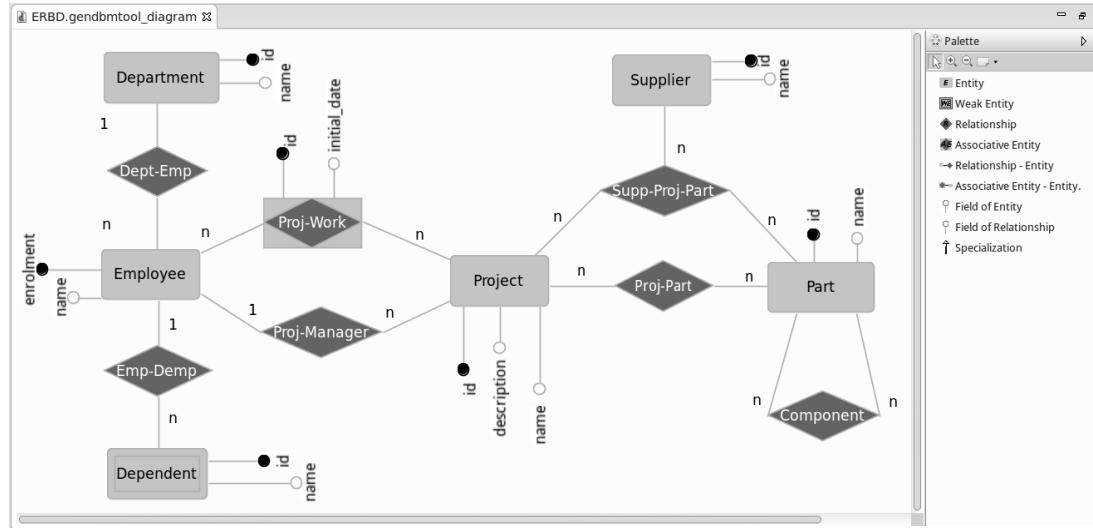


Figura 3 – Modelagem conceitual realizada com a ferramenta.

O modelo definido graficamente é armazenado em um arquivo de extensão `.gendbmttool_diagram`. Todas as definições e alterações realizadas no modelo desse arquivo também ficam registradas simultaneamente no formato de árvore hierárquica de objetos, em um arquivo de extensão `.gendbmttool`. Como a definição de modelos na interface gráfica não leva em conta aspectos do projeto lógico, através da árvore hierárquica é possível atribuir todas as integridades pertinentes ao modelo. Para isso, basta clicar com o botão direito sobre o elemento desejado e criar um elemento mais interno a partir do primeiro que represente a integridade desejada. A árvore hierárquica pode se vista na Figura 4.

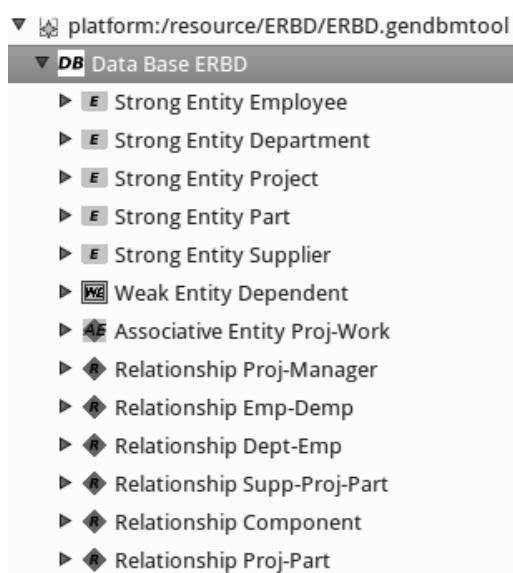


Figura 4 – Árvore hierárquica de objetos.

Para gerar o código SQL/DDL da modelagem criada é necessário ir ao diretório onde o arquivo de extensão *.gendbmtool* se encontra, clicar com o botão direito do mouse sobre o ícone do arquivo e na barra de menus que irá ser exibida ir em *GenerateCode ->SFS/SQL*. O código gerado pode ser visto na Figura 5.

```

CREATE DATABASE ERBD;
CREATE TABLE ERBD.Department (
    id int,
    name varchar(80),
    CONSTRAINT pk_department_ID PRIMARY KEY ( id )
);
CREATE TABLE ERBD.Employee (
    enrolment int,
    name varchar(80),
    id1 Department int,
    CONSTRAINT pk_employee_ID PRIMARY KEY ( enrolment ),
    CONSTRAINT fk1_department FOREIGN KEY id1_Department
        REFERENCES Department ( id )
);
CREATE TABLE ERBD.Project (
    id int,
    description varchar(80),
    name varchar(80),
    id1_Employee int,
    CONSTRAINT pk_project_ID PRIMARY KEY ( id ),
    CONSTRAINT fk1_employee FOREIGN KEY id1_Employee
        REFERENCES Employee ( enrolment )
);
CREATE TABLE ERBD.Part (
    id int,
    name varchar(80),
    CONSTRAINT pk_part_ID PRIMARY KEY ( id )
);
CREATE TABLE ERBD.Dependent (
    id int,
    name varchar(80),
    id1_Employee int,
    CONSTRAINT pk_dependent_ID PRIMARY KEY ( id ),
    CONSTRAINT fk1_employee FOREIGN KEY id1_Employee
        REFERENCES Employee ( enrolment )
);
CREATE TABLE ERBD.Proj-Work (
    id int,
    initial_date date,
    id1_Employee int,
    id2_Project int,
    CONSTRAINT pk_proj-work_ID PRIMARY KEY ( id ),
    CONSTRAINT fk1_employee FOREIGN KEY id1_Employee
        REFERENCES Employee ( enrolment ),
    CONSTRAINT fk2_project FOREIGN KEY id2_Project
        REFERENCES Project ( id )
);

```

Figura 5 – SQL gerado para a modelagem relacional.

3.2. Modelagem para Bancos de dados Geográficos

Para exemplificar o funcionamento da ferramenta em relação a modelos definidos para BDG segundo a notação OMT-G, é tido como referência um modelo extraído de [Borges et al., 2001]. O modelo utilizado retrata um banco de dados para cadastro urbano, utilizado para armazenar dados sobre o espaço geográfico do município e sobre impostos relativos ao mesmo. Esse modelo foi escolhido por utilizar vários recursos da notação OMT-G, proporcionando um exemplo mais rico em detalhes.

Para realizar a modelagem gráfica é utilizada a ferramenta OMT-G Design. Por também se tratar de uma ferramenta executada dentro do ambiente Eclipse, a interface gráfica, em relação à disposição de conteúdo, é similar a ferramenta utilizada para BDR, contendo a paleta de recursos disponíveis para a construção do modelo na barra lateral direita. A modelagem realizada através da OMT-G Design se encontra na Figura 6.

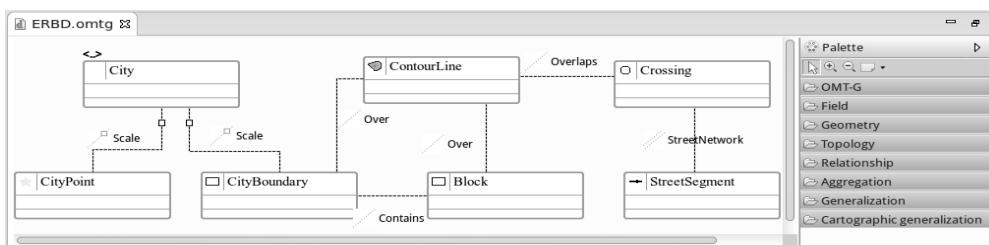


Figura 6 – Modelagem conceitual para banco de dados geográficos realizada com a ferramenta.

Depois de realizada a modelagem, o próximo passo é o mapeamento de conceitos entre o modelo OMT-G e o GEDBM. Para isso, é necessário clicar com o botão direito sobre o arquivo *.omtg* gerado e no menu que irá se abrir selecionar as opções *Run as ->Run Configuration*. Na janela que irá se abrir deve-se selecionar a opção *Operational QVT Interpreter* e a seguir clicar sobre o botão *New LaunchConfiguration*.

Com isso será aberto um novo formulário onde primeiramente deve-se selecionar o módulo de transformação a ser executado, para isso é necessário clicar sobre o botão *Browse* do campo *Transformation Module*. Na nova janela serão apresentados todos os módulos de transformação disponíveis, dentre os apresentados deve se escolher o cartucho que contém as regras de transformação de modelo para modelo e clicar em *OK*.

Tendo feito isso, dois novos campos de texto são abertos dentro do grupo de opções *Transformation parameters*. O campo abaixo do *label IN omtg Model* é onde deve ser indicado o caminho do modelo de entrada para o mapeamento, e o campo abaixo do *label OUT gendbtool Model* se indica o caminho onde o arquivo *.gendbtool* resultante do mapeamento será armazenado. Depois de feitas as devidas configurações, é necessário apenas clicar em *Run* para executar as regras de mapeamento e assim ter o arquivo *.gendbtool* gerado. Para finalmente gerar o código SFS/SQL, basta clicar com o botão direito sobre o arquivo *.gendbtool* gerado e clicar na opção *Generate Code->SFS/SQL*. Será então gerado o artefato de texto com o código desejado e que pode ser visto na Figura 7.

```

CREATE DATABASE ERBDDiagram;
CREATE TABLE ERBDDiagram.City (
    id int,
    CONSTRAINT pk_city_ID PRIMARY KEY ( id )
);
CREATE TABLE ERBDDiagram.CityBoundary (
    id int,
    CityBoundary Polygon,
    id1_City int,
    CONSTRAINT pk_cityboundary_ID PRIMARY KEY ( id ),
    CONSTRAINT fk1_city FOREIGN KEY id1_City
        REFERENCES City ( id )
);
CREATE TABLE ERBDDiagram.CityPoint (
    id int,
    CityPoint Point,
    id1_City int,
    CONSTRAINT pk_citypoint_ID PRIMARY KEY( id ),
    CONSTRAINT fk1_city FOREIGN KEY id1_City
        REFERENCES City ( id )
);
CREATE TABLE ERBDDiagram.ContourLine (
    id int,
    ContourLine LineString,
    CONSTRAINT pk_contourline_ID PRIMARY KEY ( id )
);
CREATE TABLE ERBDDiagram.Block (
    id int,
    Block Polygon,
    CONSTRAINT pk_contourline_ID PRIMARY KEY ( id )
);
CREATE TABLE ERBDDiagram.Crossing (
    id int,
    Crossing Point,
    CONSTRAINT pk_contourline_ID PRIMARY KEY ( id )
);
CREATE TABLE ERBDDiagram.StreetSegment (
    id int,
    StreetSegment LineString,
    CONSTRAINT pk_contourline_ID PRIMARY KEY ( id )
);

```

Figura 7 – Código SQL gerado pela ferramenta para o modelo geográfico apresentado.

Vale ressaltar que para os dois casos de modelagem descritos anteriormente, a partir do momento em que se possui o arquivo *.gendbtool* o processo de geração de código é o mesmo. Devido à divisão de grupos de modelos por nível de abstração que a MDA define, o *gendbtool* corresponde ao PIM a ser submetido às regras de transformação para se chegar ao PSM. Pelo fato das regras terem sido definidas para serem aderentes ao GEDBM, mesmo havendo variação na forma de modelagem para BDR e BDG, as mesmas não implicam em mudanças em todo o processo de transformação.

4. Trabalhos Relacionados

Nessa seção, são apresentadas algumas ferramentas com propósito semelhante ao da GenDBM Tool, são descritas suas principais características e traçado um comparativo entre as mesmas e a ferramenta proposta.

Com relação a ferramentas voltadas para atender BDR tem-se como exemplo a EERCASE [Fidalgo et al., 2013]. Essa ferramenta recebe como entrada modelos definidos através da notação *Enhanced Entity-Relationship Model* (EERM) por

intermédio de sua interface gráfica, realiza a validação dos modelos e executa a geração do código SQL/DDL referente ao mesmo. Para possibilitar a modelagem utilizando a notação EERM, a ferramenta conta com o meta-modelo chamado *Enhanced Entity-Relationship MetaModel* (EERMM).

A EERCASE, em relação às notações de modelagem conceitual suportadas graficamente, suporta a notação ER. A GenDBM Tool tem como objetivo ser receptiva a um número maior de notações utilizadas em SGBD relacionais além de possuir extensão para modelagem de base de dados geográfica, aspecto inexistente na EERCASE. O código SQL/DDL gerado através da EERCASE é voltado para o SGBD PostgreSQL, podendo ser editado manualmente para ser compatível com outros. A GenDBM Tool tem como base a geração de código no padrão ANSI SQL e compatível com a especificação SFS, podendo abranger vários SGBD relacionais mediante edição de algumas particularidades no código pertinentes a cada um.

No segmento de modelagem para BDG, podemos citar a ferramenta OMT-G Designe [Lizardo and Davis, 2014], que permite a definição de modelos segundo a notação OMT-G. No seu ambiente de modelagem, além dos recursos necessários para realização da modelagem conceitual, são oferecidos recursos para importação e exportação de modelos através de arquivos XML, exportação do arquivo contendo o código SQL/DDL do modelo definido e a possibilidade de imprimir o diagrama do modelo definido.

Fazendo um comparativo do OMT-G Designer com a GenDBM Tool, pode-se observar que o OMT-G Designer foi planejado exclusivamente para modelagens geográficas enquanto a ferramenta proposta, embora utilize da OMT-G Design para a notação OMT-G, possui interface para o MER. O OMT-G Designer oferece a opção de gerar o código SFS/SQL com a possibilidade de mapeamento para o formato geográfico da Oracle, enquanto a GenDBM Tool segue o padrão internacional visando ser o mais genérico possível. Pela alteração que alguns SGBD fazem com relação à sintaxe adotada da normativa do padrão ANSI SQL, em alguns casos pode ser necessária a edição manual do código para que se torne aderente ao SGBD em questão.

A GenDBM Tool não tem como objetivo substituir ou superar as ferramentas analisadas nessa seção, mas ser uma facilitadora quanto à flexibilidade de opções no momento de modelagem e quando possível trabalhar em conjunto com as mesmas.

5. Conclusão

Este trabalho apresentou a GenDBM Tool, uma ferramenta MDA para modelagem de banco de dados relacional e geográfico que permite a geração de codificação automática para o padrão ANSI SQL para banco de dados relacional; e SFS para banco de dados geográfico, a partir de diversas linguagens de modelagem e notações. A ferramenta utiliza o GEDBM, um meta-modelo genérico para modelagem conceitual como PIM, que abrange os conceitos de diversas linguagens de modelagem relacional e uma geográfica, o que permite uma flexibilização no desenvolvimento do modelo conceitual pelo projetista. O modelo instanciado, utilizando das definições do GEDBM, é transformado através de regras estabelecidas no PSM, que corresponde ao artefato de texto que contém o código gerado ANSI SQL ou SQL/SFS desejado.

Ao seu utilizar da MDA, a solução proposta pode possibilitar a integração de outras soluções ou ferramentas ao GEDBM a partir de um conjunto de transformações

entre modelos. Dessa forma, além de aproveitar os recursos de outras tecnologias, não é necessário refazer os conjuntos de regras para codificação automática, uma vez que estes já estão definidos no modelo de destino. Além disso, também é possível identificar modelagens conceituais equivalentes entre o modelo conceitual para banco de dados relacionais e geográficos, uma vez que as transformações entre modelos mapeiam todos os conceitos existentes no modelo geográfico OMT-G para o modelo GEDBM, que implementa o modelo ER graficamente.

Como trabalho futuro será necessário aplicar regras de validação em OCL para a ferramenta GenDBM Tool, de forma que modelos inconsistentes possam ser identificados antes da geração do código. Também é possível permitir a realização da engenharia reversa a partir da entrada de um *script* SQL e gerar automaticamente o modelo ER através da ferramenta Xtext.

6. Referências

- Borges, K. A. V., Davis, C. A. and Laender, A. H. F. (sep 2001). OMT-G: An Object-Oriented Data Model for Geographic Applications. *Geoinformatica*, v. 5, n. 3, p. 221–260.
- Chen, P. P.-S. (mar 1976). The entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.*, v. 1, n. 1, p. 9–36.
- Elmasri, R., Navathe, S. B., Pinheiro, M. G., et al. (2005). *Sistemas de banco de dados*. Pearson Addison Wesley.
- Fidalgo, R. N., Alves, E., Espana, S., Castro, J. and Pastor, O. (2013). Metamodeling the Enhanced Entity-Relationship Model. *Journal of Information and Data Management*, v. 4, p. 406–420.
- Guinelli, J. V., Rosa, A. S., Pantoja, C. E. and Choren, R. (2014). Uma Metodologia Para Apoio ao Projeto de Banco de Dados Geográficos Utilizando a MDA. In *X Simpósio Brasileiro de Sistemas de Informação*. . Sociedade Brasileira de Computação.
- Lizardo, L. and Davis, J., ClodoveuAugusto (2014). OMT-G Designer: A Web Tool for Modeling Geographic Databases in OMT-G. In: Indulska, M.; Purao, S.[Eds.]. . *Advances in Conceptual Modeling*. Lecture Notes in Computer Science. Springer International Publishing. v. 8823p. 228–233.
- Martinez, A. O. T. and Frozza, A. A. (2014). OMT-G Design: Uma Ferramenta para Modelagem de Dados Espaciais. *X Escola Regional de Banco de Dados*. São Francisco do Sul, SC: .
- Mellor, S. J., Scott, K., Uhl, A. and Weise, D. (2005). *MDA Destilada: Princípios de Arquitetura Orientada por Modelos*. Ciência Moderna Ltda.
- Obeo (2012). Acceleo: MDA generator - home. <http://www.acceleo.org/>.
- OMG (2003). *Model Driven Architecture (MDA) Guide*.
- Rosa, A., Gonçalves, I. and Pantoja, C. E. (2013). A MDA Approach for Database Modeling. *Lecture Notes on Software Engineering*, v. 1, n. 1, p. 26–30.

Utilização da técnica de árvore de decisão para identificação de espécies de aves do estado do Rio Grande do Sul – Brasil

Jeane Paz¹, Viviani Lopes Bastos¹, Daniel Notari², Scheila de Avila e Silva^{2,4}

¹Universidade de Caxias do Sul, Centro de Ciências Biológicas e da Saúde, Rua Francisco Getúlio Vargas, 1130, CEP 95070-560, Caxias do Sul, RS, Brasil

²Universidade de Caxias do Sul, Instituto de Biotecnologia, Rua Francisco Getúlio Vargas, 1130, CEP 95070-560, Caxias do Sul, RS, Brasil

³Universidade de Caxias do Sul, Centro de Ciências Exatas e da Tecnologia, Rua Francisco Getúlio Vargas, 1130, CEP 95070-560, Caxias do Sul, RS, Brasil

⁴Universidade de Caxias do Sul, Campus Universitário de Vacaria, Av. Dom Frei Cândido Maria Bampi, 2800, CEP 95200-000, Vacaria, RS, Brasil

{jeane_paz@yahoo.com.br, vlbastos@ucs.br, dlnotari@gmail.com, sasilva6@ucs.br}

Abstract. *The knowledge about the biodiversity of a given area is important for its conservation and management. The species classification requires specific background by the researcher in addition to the analysis of several morphological characteristics. Thus, data mining approaches can assist in this assignment by reducing the number of features which should be analyzed. Furthermore, it can be supportive for beginner researchers. The present paper describes the application of decision trees in the species classification. In total, 213 records of species, each one with 15 visual attributes have been used in our experiments. The attributes ventral stripe color, head color, belly underside of wings and details of tail were those used to build the decision tree.*

Resumo. *A conservação e manejo de uma área relacionam-se com o conhecimento da biodiversidade local. A identificação de uma espécie requer conhecimento prévio por parte do pesquisador, além da análise de múltiplas características morfológicas. A mineração de dados contribui neste processo reduzindo o número de características a serem analisadas e/ou auxiliar pesquisadores principiantes. O presente trabalho propõe a aplicação da técnica de árvores de decisão na classificação das espécies de Aves da ordem Falconiformes. Foram utilizados 213 exemplos e 15 atributos visuais, sendo os que apresentaram maior capacidade de classificação foram: cor das listras ventrais, cor da cabeça, parte inferior da asa e detalhes da cauda.*

1. Introdução

A análise do volume de dados disponíveis, nas diversas áreas do conhecimento, possibilita a extração de informações para a fundamentação de novas inferências e contribuições a partir de dados já existentes e disponíveis. A Biologia também se insere neste cenário, uma vez que a quantidade de dados gerados experimentalmente pode servir como fonte para novas inferências, por meio da combinação de dados e aplicação de técnicas computacionais para descoberta de informações [MARX, 2013]. Este tipo de

pesquisa está transformando áreas clássicas, como a taxonomia, a biologia comparativa, a biologia molecular, a genética, entre outras [MARX, 2013]. A mineração de dados é uma alternativa eficaz para extrair informações a partir de grandes volumes de dados, descobrindo relações ocultas, padrões e gerando regras para predizer e correlacionar dados. O resultado destas análises podem tornar mais rápido o processo de tomada de decisão ou proporcionar um maior grau de confiança. A principal vantagem desta abordagem é que não são necessárias hipóteses, sendo que o conhecimento é extraído dos dados sem conhecimento prévio [GALVÃO e MARIN, 2009].

O Brasil é um país com elevado índice de biodiversidade e, paralelamente, também apresenta altos índices de espécies ameaçadas de extinção [FERREIRA et al., 2005]. Com a crescente expansão da população mundial, as ações antrópicas tornaram-se intensas, afetando a dinâmica dos ecossistemas [BORSATO et al., 2004]. A conservação e manejo adequado de uma determinada área é dependente do conhecimento a respeito da diversidade de espécies e suas características [MARINI e GARCIA, 2005]. Neste contexto, as aves são importantes na avaliação da qualidade ambiental e na determinação de áreas para a conservação. Isto se deve ao fato destas ocuparem diferentes habitats, níveis tróficos e apresentarem sensibilidade às modificações ambientais [VALADÃO, 2013]. Dentre os seres vivos de um ecossistema, as aves atuam não só na polinização de flores [MARTINS e BATALHA, 2006] como também na dispersão de sementes. Certos frutos, como por exemplo o da aroeira vermelha, aumentam a porcentagem e velocidade de germinação após suas sementes passarem pelo trato digestivo de alguma ave [D'AVILA et al., 2010; GUERTA et al., 2011]. Adicionalmente, elas também exercem o papel de controle biológico na população de alguns invertebrados e até mesmo de outros vertebrados, além de serem fonte de alimento para outros predadores [SICK, 1997]. Apesar de as aves serem um grupo amplamente estudado, conhecer e sistematizar sua biologia é um assunto ainda relevante e com muitas questões a serem respondidas, uma vez que informações sobre certas espécies são escassas ou ausentes [MAGALHÃES et al., 2007].

Na área da taxonomia, a mineração de dados pode ser utilizada como um facilitador da classificação de espécies. Esta tarefa geralmente exige paciência, bem como o porte de guias e chaves de classificação eficientes. Neste caso, a presença ou ausência de conhecimento prévio de taxonomia por parte do pesquisador influencia diretamente a eficiência e eficácia do trabalho realizado [CAVARZERE et al., 2013]. Assim, a mineração de dados contribui no trabalho do pesquisador uma vez que a classificação resultante provém da análise de atributos que possuem características mais discriminatórias para um grupo específico. Deste modo, o trabalho dos pesquisadores pode ser otimizado, uma vez que a fonte de pesquisa é reduzida [GALVÃO e MARIN, 2009]. Considerando este contexto, o presente artigo utilizou a técnica de aprendizagem de máquina de árvores de decisão para a realização de mineração de dados característicos de aves Falconiformes do estado do Rio Grande do Sul. Assim, pretende-se contribuir para a identificação destas aves em campo por meio da extração de informações que possam auxiliar na redução da utilização de chaves complexas de classificação em trabalho de campo.

2. Trabalhos Relacionados

As técnicas de mineração de dados vêm sendo aplicadas em áreas como medicina, finanças, comércio, marketing, telecomunicações, meteorologia, agropecuária, bioinformáticas, entre outras. Até o momento de conclusão deste artigo, não foram

encontrados artigos que tratavam da aplicação de árvores de decisão especificadamente em aves falconiformes. Assim, os trabalhos relacionados tratam de mineração de dados relacionados com as áreas biológicas e da saúde. Na área médica, a aplicação das técnicas de mineração apresentam inúmeros exemplos, como no controle de infecções hospitalares (DAO et al., 2008), diminuição de erros em diagnósticos de hipertensão (DÁVILA HERNÁNDEZ e CORALES, 2012), determinação de variáveis psicossociais que influenciam no consumo de nicotina em adolescentes (MONTAÑO-MORENO et al., 2014), entre outros.

Além destes trabalhos, STEINER et. al (2006) mostram uma aplicação de no diagnóstico diferencial da icterícia, comparando as técnicas de: Programação Linear, Função Discriminante Linear de Fisher, Modelo de Regressão Logística, Árvores de Decisão e Redes Neurais Artificiais com dois conjuntos de dados: um com os dados originais e outro conjunto com a eliminação de seis dos atributos originais. Os resultados com maior exatidão foram obtidos com o segundo conjunto de dados, os quais variaram entre 75% com as árvores de decisão até 96% com as Redes Neurais Artificiais e Programação Linear. Assim, os autores ilustram o potencial das técnicas de mineração de dados como ferramenta auxiliar para a tomada de decisão em tarefas desempenhadas por especialistas. Por outro lado, VIANNA et al. (2010) utilizaram árvores de decisão em dados provenientes da integração de dados de três diferentes sistemas de informação: (i) Sistema de Informações sobre Mortalidade (SIM); (ii) Sistema de Informações sobre Nascidos Vivos (SINASC) e (iii) do Sistema de Investigação da Mortalidade Infantil (SIMI) para obter o perfil da mortalidade infantil no Estado do Paraná, no período de 2000 a 2004. As 4.230 regras obtidas foram analisadas por 22 especialistas dos Comitês Regionais de Saúde do estado do Paraná que selecionaram quatro regras de maior relevância e frequência e com menor taxa de erro. Nesta análise, destaca-se que 55% dos óbitos poderiam ser evitáveis se houvesse uma adequada atenção à gestação, parto e ao recém-nascido. O sexo predominante para óbitos foi o masculino (58%) e a idade prevalente da mãe ficou no intervalo de 16-35 anos (87%).

Na área de bioinformática, há trabalhos sobre análise da expressão gênica (TORRES-AVILÉS, 2014), as árvores de decisão foram utilizadas por DE AVILA e SILVA et al. (2011). Neste trabalho, os autores estabeleceram perfis para as sequências promotoras da bactéria *Escherichia coli* por meio da comparação das regras obtidas das simulações de redes neurais com as obtidas com as árvores de decisão. Assim, contribuindo para o processo *in silico* de reconhecimento de sequências reguladoras. Na área ambiental, TSAI et al. (2013) utilizaram árvores de decisão e redes bayesianas para extrair informações sobre em deslizamentos induzidos por chuvas fortes e relacionaram os dados. Para isso, os autores utilizaram 11 atributos formados por fatores topográficos e vegetativos. As regras criadas foram aplicadas para predizer potenciais regiões com risco de deslizamentos, mostrando uma acurácia geral de 95% tanto para os resultados com as árvores de decisão, quanto com as redes bayesianas. Além deste trabalho, BOSCHI et al. (2011), analisaram o comportamento espaço-temporal da precipitação pluvial no Estado do Rio Grande do Sul, entre os decênios de 1987-1996 e 1997-2006, por meio de técnicas de mineração de dados. Outro exemplo, de utilização de árvores de decisão na análise de cobertura vegetal de solo é apresentado por LATORRE et al., 2007, o qual aplicaram a metodologia na análise de área florestal amazônica.

Na área da agricultura e pecuária, um exemplo da utilização das árvores de decisão na compreensão de manifestações epidêmicas da ferrugem do cafeeiro. Para realizar a classificação da taxa de infecção, foram utilizados atributos meteorológicos,

carga de frutos do cafeiro e espaçamento entre as plantas de 264 exemplos retirados de lavouras no período de 1998 a 2006. A árvore apresentou uma exatidão de 73% para novos exemplos, sendo os atributos mais significativos para a classificação foram a temperatura média no período de molhamento foliar, carga pendente dos frutos, média da temperatura máxima diária no período de incubação e umidade relativa do ar. Além deste trabalho, MAIA et al. (2013), utilizaram as árvores de decisão para avaliar o conforto termal de cavalos, a qual apresentou 74 % de exatidão e 6 regras importantes para os especialistas de domínio.

3. Metodologia

O presente trabalho utilizou uma abordagem computacional para a obtenção dos objetivos propostos. Assim, a metodologia utilizada foi um subconjunto das etapas de um sistema de *Business Intelligence* com ênfase em dados biológicos, sendo que esta metodologia pode ser generalizada para outras espécies biológicas. As quatro etapas principais realizadas estão ilustradas na Figura 1.

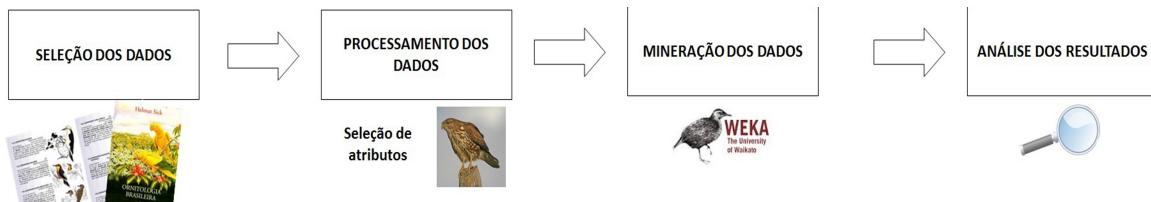


Figura 1. Etapas da metodologia adotada

A quantidade de informação sobre as características de cada espécie de aves é rica em livros e artigos científicos, porém muito pobre e carente de fontes confiáveis na internet. Foram utilizados então, como base para obtenção de dados, guias de campo e livros sobre ornitologia [SICK, 1997; NAROSKY e YZURIETA, 2003]. Assim, este trabalho aponta para a necessidade da criação de um banco de dados com acessibilidade pela internet para a realização de pesquisas relacionadas à computação aplicada, uma vez que foi necessária a construção de uma base de dados que foram utilizados na etapa 1.

Na etapa 1, os dados foram coletados no período fevereiro de 2013 a novembro de 2013. Eles constituíram-se fundamentalmente de Aves (Reino *Animalia*) da ordem Falconiformes e todas suas respectivas famílias, totalizando 213 exemplos. A escolha pela análise dos Falconiformes justifica-se pelo de fato desta ordem ter sua diversidade sensível às perturbações que geram a perda do habitat [PETERSEN et al., 2011]. Além disso, a falta de informações sobre o tamanho populacional das espécies e da tolerância aos distúrbios ambientais são fatores que dificultam o estabelecimento de programas prioritários para a proteção das aves de rapina [RODA e PEREIRA, 2006]. Deste modo, uma ferramenta auxiliar na classificação biológica é importante.

A segunda etapa consistiu na separação dos dados em atributos, que consistiram em: (i) tamanho (altura em centímetros); (ii) cor da vista ventral; (iii) presença de listras ventrais; (iv) cor das listras ventrais; (v) cor da vista dorsal; (vi) cor da cabeça; (vii) descrição de detalhes na nuca; (viii) descrição de detalhes na garganta; (ix) cor dos calções; (x) descrição da parte inferior da asa; (xi) descrição de detalhes da cauda (sendo este dividido em dois atributos pela presença de múltiplas características); (xii) descrição de detalhes nos ombros; (xiii) descrição de detalhes no traseiro; (xiv) descrição de detalhes do peito. Estes 15 atributos foram escolhidos devido ao fato de a identificação em campo ser realizada por meio da observação de características externas, utilizando registros fotográficos, ou com a captura de espécies para avaliação de dados mais

específicos como altura, peso e sexo. Para algumas espécies nem todos os atributos possuíam valores, sendo nestes casos utilizada a expressão “Ausente”.

Na etapa 3, a mineração de dados foi realizada utilizando a ferramenta WEKA. Para alcançar os objetivos propostos, o algoritmo escolhido foi o J48, o qual é uma implementação em Java do algoritmo C4.5 [WITTEN et al., 2011]. Uma árvore de decisão tem a função de dividir um conjunto de treinamento, até que cada subconjunto obtido deste particionamento contenha casos de uma única classe [LATORRE et al., 2007]. Uma árvore de decisão é representada por nodos que simbolizam os atributos, sendo estes conectados por arcos. Os arcos são provenientes dos nodos e recebem os valores possíveis para cada atributo. O primeiro nodo é chamado raiz, pois dele derivam os outros nodos, chamados de folha, que representam o conjunto das diferentes classes de um conjunto de treinamento. Assim, esta estrutura hierárquica de nós internos é responsável pela tomada de decisão. O último nodo representa o objetivo da árvore, sendo que estes não possuem nó descendente. No caso deste trabalho, o nodo final era a espécie de cada ave. A divisão em cada nó interno da árvore baseia-se no critério de ganho de informação obtido na escolha do atributo para subdivisão [WITTEN et al., 2011].

Para a obtenção de dados estatisticamente válidos, optou-se pela metodologia *k-fold cross-validation*. Esta técnica consiste em particionar os dados de entrada em *k* partes iguais e realizar o treinamento com *k-1* partes e testar a classificação com a *k* parte faltante [DE AVILA E SILVA et al., 2011]. Este processo de treino e validação é realizado *k* vezes, sendo em cada repetição utilizado um conjunto de treino e teste diferente. Dentre os parâmetros utilizados, destaca-se que o fator de confidência foi de 0,25 e o número mínimo de padrões classificados por folha foi 10. Assim, foi possível gerar uma árvore com regras mais generalistas. A aplicabilidade das regras biológicas geradas com a árvore de decisão foram validadas por um especialista de domínio e consulta de literatura específica.

4. Resultados e Discussão

Considerando que o levantamento e taxonomia de espécies necessita de uma vasta quantidade de dados à disponibilidade do pesquisador, o principal objetivo deste trabalho foi implementar uma solução auxiliar para a identificação de espécies. Com os dados fornecidos, foi gerada uma árvore de 102 folhas, a qual é apresentada de forma parcial na Figura 2. Optou-se por remover regras redundantes a fim de facilitar a análise e discussão dos resultados. O tamanho da árvore é um indicativo do alto grau de entropia dos dados, uma característica dos dados biológicos [MARX, 2013]. Apesar desta característica entrópica, a árvore de decisão gerada classificou corretamente 89% dos exemplos apresentados e um erro médio quadrático de aproximadamente 6%.

A árvore de decisão apresentada na Figura 2, mostra que o nodo raiz é formado pelo atributo “cor das listras ventrais”. Assim, apenas utilizando a cor das listras ventrais é possível identificar a espécie *Accipiter bicolor*. Caso não haja listras ventrais, a segunda informação que deve ser analisada é a cor da cabeça (Figura 2). Com base na árvore de decisão obtida, foi possível descrever regras do tipo SE-ENTÃO com validação na literatura relacionada, as quais são apresentadas na Figura 3.

As regras apresentadas na Figura 3 apresentaram validação na literatura disponível da área, sendo que a regra (a) está de acordo com os dois guias para identificação de aves em campo [NAROSKY e YZURIETA, 2003; SICK, 1997].

```

Cor das listras = Marrom e branco: Accipiter bicolor
Cor das listras = Ausente
| Cabeça = Cinza: Leptodon cayanensis
| Cabeça = Negra
| | Parte inferior da asa = Marrom: Buteo albicaudatus
| | Parte inferior da asa = Bordas negras: Buteo albicaudatus
| | Parte inferior da asa = Bordas listradas: Buteo brachyurus
| | Parte inferior da asa = Bordas brancas: Buteo albicaudatus
| | Parte inferior da asa = Pontas brancas: Coragyps atratus
| | Parte inferior da asa = Listras brancas: Buteo albicaudatus
| | Parte inferior da asa = Alaranjada: Buteo albicaudatus
| | Parte inferior da asa = Mancha branca: Buteo albicaudatus
| | Parte inferior da asa = Negra: Rostrhamus sociabilis (macho)
| Cabeça = Branca: Busarellus nigricollis
| Cabeça = Marrom: Harpia harpyja
| Cabeça = Negra parda: Buteo swainsoni
| Cabeça = Vermelha: Cathartes aura
| Cabeça = Amarela/Rosada: Cathartes burrovianus
| Cabeça = Negra/face branca: Circus buffoni
| Cabeça = Cinza/marrom: Harpia harpyja
| Cabeça = Cinzenta/listras: Harpia harpyja
| Cabeça = Supercílio branco: Harpia harpyja
| Cabeça = Colorida: Sarcophagus papa
| Cabeça = Cinza escura: Spizapteryx circumcinctus
Cor das listras = Marrom
| Vista ventral = Branca: Accipiter striatus
| Vista ventral = Marrom: Accipiter striatus
| Vista ventral = Alaranjada: Heterospizias meridionalis
| Vista ventral = Amarelada: Rostrhamus sociabilis (fêmea)
Cor das listras = Alaranjadas
| Vista dorsal = Negra: Harpyhaliaetus coronatus
| Vista dorsal = Marrom: Buteo magnirostris
| Vista dorsal = Cinza/marrom: Circus cinereus
Cor das listras = Brancas
| Garganta = Ausente
| | Vista dorsal = Negra: Spizaetus tyrannus
| | Vista dorsal = Marrom: Milvago chimango
| | Vista dorsal = Cinza escuro: Falco femoralis
Cor das listras = Negras
| Cabeça = Cinza: Falco sparverius
| Cabeça = Negra: Falco peregrinus
| Cabeça = Branca: Milvago chimachima
| Cabeça = Marrom: Micrastur ruficollis
| Cabeça = Cinza escuro: Geranoaetus melanoleucus
| Cabeça = Alaranjada: Spizaetus ornatus
Cor das listras = Cinza escuro: Ictinia plumbea
Cor das listras = Marrons/negras: Morphnus guianensis
Cor das listras = Amareladas: Parabuteo unicinctus

```

Figura 2. Árvore de Decisão gerada com os dados selecionados

A regra (b) está em conformidade com a descrição de JOENCK e AZEVEDO (2006), os quais avistaram um exemplar desta espécie com a seguinte morfologia: coloração da plumagem clara/branca no mento, na região ventral e lateral do pescoço (com algumas esparsas penas estriadas de coloração marrom-claro), coloração acinzentada no dorso, (incluindo as asas) e na cabeça (da fronte até a nuca). A regra (c) possui descrição compatível com o trabalho de SILVA e OLIMOS (2007), que avistaram esta espécie, descrevendo-a como morfo-escura.

Outra informação em conformidade com a literatura foi a diferença entre macho e fêmea da espécie *Rostrhamus sociabilis*, conforme regas (d) e (e). A regra (f), a qual classifica a espécie *Busarellus nigricollis*, apresenta as características analisadas por AMARAL (2002) na identificação e diferenciação desta espécie de outras pertencentes à família *Accipitridae*. Neste caso, ressalta-se a coloração bege escura e ferrugínea do corpo, cabeça branca, com uma mancha negra no papo e pela cauda extremamente curta.

- (a) SE cor das listras ventrais for marrom e branco ENTÃO a espécie é *Accipiter bicolor*.
- (b) SE cor das listras ventrais for ausente E a cabeça for cinza ENTÃO a espécie é *Leptodon cayanensis*.
- (c) SE a cor das listras ventrais for ausente E a cabeça negra E a parte inferior da asa com bordas listradas ENTÃO a espécie é *Buteo brachyurus*.
- (d) SE a cor das listras ventrais for ausente E a cabeça e a parte inferior da asa negras ENTÃO se trata de um indivíduo macho da espécie *Rostrhamus sociabilis*.
- (e) SE a cor das listras ventrais for ausente e a vista ventral for amarelada ENTÃO se trata de uma fêmea da espécie *Rostrhamus sociabilis*.
- (f) SE a cor das listras ventrais for ausente e a cabeça branca ENTÃO a espécie é *Busarellus nigricollis*.

Figura 3. Regras SE-ENTÃO obtidas a partir da Árvore de Decisão com validação na literatura

Outras regras validadas pelo especialista de domínio e pelos guias de campo estão apresentadas na Figura 4. No entanto, não foram encontrados relacionados para a validação das regras até o momento da conclusão deste artigo.

- (a) SE a cor das listras ventrais for ausente e a cabeça negra com a face branca ENTÃO a espécie é *Circus buffoni*.
- (b) SE a cor das listras ventrais for ausente e a cabeça cinza escura ENTÃO a espécie é *Spizapteryx circumcinctus*.
- (c) SE a cor das listras ventrais for branca, o detalhe da garganta ausente e a vista dorsal for cinza escuro ENTÃO a espécie é *Falco femoralis*.
- (d) SE a cor das listrais ventrais for negra e a cabeça cinza ENTÃO a espécie é *Falco sparverius*.
- (e) SE a cor das listrais ventrais for negra e a cabeça negra ENTÃO a espécie é *Falco peregrinus*.
- (f) SE a cor das listrais ventrais for negra e a cabeça alaranjada ENTÃO a espécie é *Spizaetus ornatus*.
- (g) SE a cor das listras ventrais for marrom e negra ENTÃO a espécie é *Morphnus guianensis*.
- (h) SE a cor das listras ventrais for amarelada ENTÃO a espécie é *Parabuteo unicinctus*.

Figura 4. Regras SE-ENTÃO obtidas a partir da Árvore de Decisão com validação de especialista de domínio

Outras informações obtidas a partir da Figura 2 referem-se às espécies *Buteo albicaudatus* e *Harpia harpyja*. Nestes casos, observa-se que não foi possível ter uma regra específica que os destacassem das demais, ou seja, há muitas regras diferentes para a mesma espécie. A classificação destas espécies é baseada em um mesmo atributo com diferentes descrições, sendo esta divergente com a literatura específica da área [NAROSKY e YZURIETA, 2003; SICK, 1997]. Também é possível verificar na Figura 2 que alguns atributos são restritos a um grupo de espécies. Por exemplo, a cor das listras ventrais marrom abrange 3 espécies (*Accipiter striatus*, *Heterospizias meridionalis*, *Rostrhamus sociabilis* – fêmea); o mesmo sendo percebido no caso da cor das listras ventrais alaranjadas (*Harpyhaliaeetus coronatus*, *Buteo magnirostris*, *Circus cinereus*). A cor das listras brancas, associadas com o detalhe da garganta ausente, também se limitou a 3 espécies específicas (*Spizaetus tyrannus*, *Milvago chimango*, *Falco femoralis*).

Apesar de terem sido analisados 15 atributos, nem todos foram considerados para gerar a árvore de decisão. Os atributos com maiores ganhos de informação foram: cor das listras ventrais, cor da cabeça, parte inferior da asa, detalhes da cauda, vista ventral, vista dorsal e detalhe da garganta. Por outro lado, atributos importantes na identificação de aves em campo, por exemplo, o tamanho da ave, não foi um atributo com ganho de informação suficiente para ser incluído na árvore de decisão. Assim, após a seleção de atributos pela técnica de árvores de decisão, a próxima etapa seria a aplicação das regras obtidas por um especialista durante uma atividade de campo.

5. Considerações Finais

A mineração de dados na biologia é um desafio, uma vez que os dados são ruidosos e há necessidade de combinar informações disponíveis em diferentes bases de dados [MARX, 2013]. O problema inicial de aplicação da metodologia foi a falta de repositórios *on-line* para levantamento dos dados. Não existe um repositório oficial, organizado e acurado, que forneça características de espécies de aves do Rio Grande do Sul ou até do Brasil. Assim, ressalta-se a importância da existência de um banco de dados para acesso aos dados, de modo que facilite a pesquisa aplicada. Além disso, a aplicação de técnicas computacionais para a resolução de problemas em outras áreas mostra-se como uma importante contribuição para a área de computação, uma vez que a multidisciplinaridade é uma característica de áreas relacionadas à tecnologia da informação.

Os atributos das aves que apresentaram maior capacidade de classificação, hierarquicamente, foram: cor das listras ventrais, cor da cabeça, parte inferior da asa e os detalhes da cauda. Pode-se também observar a cor das listras, a vista ventral ou dorsal, ou os detalhes da garganta. Como não foi possível encontrar artigos para validação de todas as regras obtidas, a próxima etapa deste trabalho será a utilização destas regras para realização de um estudo a campo para afirmar sua eficácia.

O processo de mineração de dados pode contribuir à área da biologia auxiliando na identificação dos padrões de espécies. Contudo, vale destacar que dependendo do objetivo proposto devem-se aplicar tarefas e métodos específicos. Esta ferramenta pode ser aplicada na taxonomia de espécies de qualquer reino, desde que se obtenham dados suficientes que diferenciem as espécies e permitam uma boa acurácia nos resultados. A mineração de dados mostra-se como uma ferramenta auxiliar ao pesquisador, no entanto, não substitui a necessidade do pesquisador e profundo domínio de exploração.

References

- Amaral, C. Ocorrência do gavião-belo *Busarellus nigricollis* no estado de Santa Catarina. Ararajuba, v. 10, n.2, p.245-245, 2002.
- Borsato, V. A.; Souza-filho, E. E. Ação antrópica, alterações nos geossistemas, variabilidade climática: contribuição do problema. Revista Formação, v. 2, n. 11, p. 213-223, 2004.
- Boschi, R. S; Oliveira, S. R. de M; Assad, E. D. Técnicas de mineração de dados para análise da precipitação pluvial decenal no Rio Grande do Sul. Eng. Agríc., v. 31, n. 6, 2011.
- Cavarzere, V.; Alves, F.; et al. Evaluation of methodological protocols using point counts and mist nets: a case study in southeastern Brazil. Pap. Avulsos Zool., v. 53, n. 26, 2013.
- Dao, T. K.; Zabaneh, F.; Holmes, J.; et al. A radical data mining method to link hospital microbiology and an infection control database. AJIC, v. 36, n. 3, 2008.
- Dávila Hernandez, F.; SANCHEZ CORALES, Yovannys. Técnicas de minería de datos aplicadas al diagnóstico de entidades clínicas. RCIM, Ciudad de la Habana, v. 4, n. 2, 2012.
- D'Avila, G.; Gomes-Jr., A.; Canary, A. C.; Bugoni, L. The role of avian frugivores on germination and potential seed dispersal of the Brazilian Pepper *Schinus terebinthifolius*. Biota Neotrop., v. 10, n. 3, 2010.
- de Avila e Silva, S.; Gerhardt, G. J.L.; Echeverrigaray, S. Rules extraction from neural networks applied to the prediction and recognition of prokaryotic promoters. Genet. Mol. Biol., v. 34, n. 2, 2011.
- Ferreira, R. C.; Machado, A. A.; Caxambu, M. G.; Ide, A. L. Levantamento de espécies de aves e das espécies vegetais forrageadas na estação ecológica do cerrado em Campo Mourão – PR. Atualidades ornitológicas, n. 127, p. 28, 2005.
- Guerta, R.S.; Lucon, L. G.; Motta-junior, J. C.; Vasconcellos, L. A. S.; Bird frugivory and seed germination of *Myrsine umbellata* and *Myrsine lancifolia* (Myrsinaceae) seeds in a cerrado fragment in southeastern Brazil. Biota Neotrop., v. 11, n. 4, 2011.
- Galvão, N. D.; Marin, H. F. Técnica de mineração de dados: uma revisão da literatura. Acta Paul Enferm, v. 22, n. 5, 2009.
- Joenck, C. M.; Azevedo, M. A. G. Novos registros de *Leptodon cayanensis* (Accipitridae) no Rio Grande do Sul e Santa Catarina, Brasil [New records of *Leptodon cayanensis* in Rio Grande do Sul and Santa Catarina, Brazil]. Revista Brasileira de Ornitologia, v.14. p. 423-425, 2006.
- Latorre, M. L.; Carvalho, J. R. O. A.; Santos, J. R. Integração de dados de sensoriamento remoto multi resoluções para a representação da cobertura da terra utilizando campos contínuos de vegetação e classificação por árvores de decisão. Revista Brasileira de Geofísica, v. 25, n. 1, p. 63-74, 2007.
- Magalhães, V. S. et al. Biologia de aves capturadas em um fragmento de Mata Atlântica, Brasil. Rev. Bras. Zool., v. 24, n. 4, 2007.
- Maia, A. P. de A. et al. A decision-tree-based model for evaluating the thermal comfort of horses. Sci. Agric., v. 70, n. 6, 2013.

- Martins, F. Q.; Batalha, M. A. Pollination systems and floral traits in cerrado woody species of the upper Taquari region (central Brasil). *Brazilian Journal of Biology*, v. 66, n. 2A, p. 543-552, 2006.
- Marx, V. Biology: The big challenges of big data. *Nature*, v.498, p. 255–260, 2013.
- Montaño-Moreno, J. J., Gervilla-García, E., Cajal-Blasco, B. et al. Data mining classification techniques: an application to tobacco consumption in teenagers. *Anal. Psicol.*, 2014, vol.30, no.2, p.633-641. ISSN 0212-9728.
- Narosky, T.; Yzurieta, D. Guía para la identificación de las aves de Argentina y Uruguay, Buenos Aires: Vazquez Mazzini, 2003. 346 p.
- Petersen, E. S.; Petry, M. V.; Krüger-Garcia, L. *Revista Brasileira de Ornitologia*, v. 19, n. 3, p. 376-384, 2011.
- Roda, S. A.; Pereira, G. A. Distribuição recente e conservação das aves de rapina florestais do Centro Pernambuco. *Revista Brasileira de Ornitologia*, v. 14, n. 4, p. 331-344, 2006.
- Sick, H.; Pacheco, J. F. *Ornitologia brasileira*. Rio de Janeiro: Nova Fronteira. p. 909, 1997.
- Silva e Silva, R.; Olmos, F. Adendas e registros significativos para a avifauna dos manguezais de Santos e Cubatão, SP. *Revista Brasileira de Ornitologia*, v. 15, n. 4, p. 551-560, 2007.
- Steiner, M. T. A. et al. Abordagem de um problema médico por meio do processo de KDD com ênfase à análise exploratória dos dados. *Gest. Prod.*, v. 13, n. 2, 2006.
- Torres-Avilés, F., Romeo, J. S., López-Kleine, L. Data mining and influential analysis of gene expression data for plant resistance gene identification in tomato (*Solanum lycopersicum*). *Electronic Journal of Biotechnology*, n. 17, p.79-82, 2014.
- Tsai, F.; Lai, J. S.; Chen, W. W.; Lin, T. H. Analysis of topographic and vegetative factors with data miningfor landslide verification. *Ecological Engineering*, n. 61, p. 669-677, 2013
- Valadão, R. M. As aves da Estação Ecológica Serra das Araras, Mato Grosso, Brasil. *Biota Neotrop.*, v. 12, n. 3, 2012.
- Vianna, R. C. X. F. et al. Mineração de dados e características da mortalidade infantil. *Cad. Saúde Pública*, v. 26, n. 3, 2010.
- Witten I. H.; Eibe F.; Hall, M.A. *Data Mining, practical Machine Learning Tools and Techniques*. 3 ed. Morgan Kaufman Publishers, 2011.

Sistemas de Recomendação e Computação Ubíqua: Um Survey

Igor Eduardo Viana Rudel¹, Juçara Salete Gubiani¹, Daniel Lichtenow¹

¹Colégio Politécnico – Universidade Federal de Santa Maria (UFSM)
Av. Roraima, nº1000, Campus UFSM – 97.105-900– Santa Maria – RS – Brasil
xksoberbado@gmail.com, jucara@ufsm.br, dlichtnow@politecnico.ufsm.br

Abstract. This work presents some possibilities generated by the application of Ubiquitous Computing technologies in Recommender Systems. Initially, the main concepts related to these areas are introduced. After, examples of Recommender Systems that use Ubiquitous Computing technologies are presented. Finally, prospects and challenges are discussed.

Resumo. Este trabalho apresenta algumas das possibilidades geradas pela integração entre as tecnologias relacionadas à Computação Ubíqua e os Sistemas de Recomendação. Inicialmente é feita uma breve caracterização destas duas áreas. Após, são apresentados Sistemas de Recomendação que utilizam, em algum grau, os recursos relacionados à Computação Ubíqua. Finalmente são discutidas perspectivas e desafios.

1. Introdução

Sistemas de Recomendação (SRs) procuram auxiliar na identificação de itens que possam ser úteis para seus usuários [Adomavicius e Tuzhilin 2005]. Nos SRs, estes itens podem ser artigos, filmes, músicas, livros, restaurantes, atrações turísticas, trilhas de esqui, dentre outros. Há mais de uma década, pesquisadores salientaram o fato de que os SRs podem ser úteis no mundo físico como vem sendo no mundo digital, “preenchendo um importante *gap* na Computação Ubíqua” [McDonald 2003]. A partir disto, o objetivo principal deste trabalho é apresentar algumas das possibilidades oriundas da integração entre Computação Ubíqua e SRs. Como [Bobadilla 2013], [Mettouris Papadopoulos 2014], [Ricci 2010] e [Felfernig 2013] o trabalho é caracterizado como um *survey*. Neste sentido, aborda alguns dos aspectos citados nestes estudos e diferencia-se deles, por dar maior ênfase à obtenção de dados sobre contexto e também por identificar trabalhos desenvolvidos por pesquisadores brasileiros.

O artigo começa na seção 2 apresentando as principais abordagens utilizadas nos SRs e discute questões relacionadas à contextualização da recomendação e aquisição de dados. A seção 3 apresenta definições sobre Computação Ubíqua. A seção 4 descreve alguns SRs que utilizam recursos relacionados à Computação Ubíqua. A partir do exposto na seção 4, na seção 5 é feita uma breve análise do cenário apontando alguns desafios a serem considerados no desenvolvimento de SRs Ubíquos. Finalmente, na seção 6 são apresentadas as considerações finais.

2. Sistemas de Recomendação

Sistemas de Recomendação (SRs) auxiliam na identificação de itens que possam ser úteis para seus usuários [Adomavicius e Tuzhilin 2005]. Um grande número de autores considera três abordagens básicas para SRs: Baseada em Conteúdo, Colaborativa e Híbrida [Adomavicius e Tuzhilin 2005]. Alguns autores apresentam outras abordagens, como a Demográfica e a Baseada em Conhecimento.

Na abordagem Baseada em Conteúdo são recomendados itens que tenham similaridade com itens que o usuário gostou no passado. Na abordagem Colaborativa são recomendados itens que foram bem avaliados por pessoas que tenham gosto similar aos do usuário alvo. Na abordagem Demográfica são levados em conta atributos como sexo, idade, por exemplo [Pazzani 1999]. Na abordagem Baseada em Conhecimento, o conhecimento do domínio assume importância maior, sendo a preferência do usuário expressa por meio de regras que definem atributos que os itens devem possuir (por exemplo “o preço do item não pode ser maior do que X”). Já a abordagem Híbrida consiste na combinação de diferentes abordagens, tentando desta forma, minimizar os problemas inerentes a cada uma delas. Informações mais detalhadas sobre as abordagens podem ser vistas em [Adomavicius e Tuzhilin 2005].

2.1. O Contexto em Sistemas de Recomendação

Independentemente da abordagem utilizada para gerar recomendações, dentre os temas que vem recebendo a atenção dos pesquisadores da área de SRs, está a necessidade de observar o contexto no qual uma recomendação é realizada. Contexto é frequentemente definido como “qualquer informação que pode ser usada para caracterizar a situação de uma entidade (uma pessoa, um lugar ou um objeto) que é considerada relevante para a interação entre o usuário e uma aplicação, incluindo o próprio usuário e a própria aplicação” [Dey 2001].

SRs que levam em conta o contexto são referenciados atualmente como Sistemas de Recomendação Conscientes do Contexto (*Context-Aware Recommender Systems – CARS*) [Adomavicius e Tuzhilin 2011]. A preocupação com o contexto nos SRs não é nova, sendo identificada, por exemplo, em [Herlocker e Konstan 2001]. Dentre os primeiros trabalhos que abordaram a questão do contexto em SRs está [Adomavivius et al. 2005] que propõe uma abordagem multidimensional baseada no modelo de dados utilizados em *Data Warehousing* e *On-Line Analytical Processing – OLAP*. Seguindo este modelo multidimensional, a nota dada a um filme, por exemplo, seria resultado da avaliação feita por um usuário, para um filme visto (ou a ser visto) com um determinado tipo de companhia (filhos, pais, namorada, etc.). Outro trabalho precursor é [Chen 2005], onde técnicas de Filtragem Colaborativa são estendidas, sendo proposta a construção de um SR turístico para telefones celulares onde os dados que caracterizam o contexto (temperatura, localização, por exemplo) são capturados mediante aplicações e sensores embutidos em telefones. A captura de dados que permitem caracterizar o contexto é discutida na seção 2.2.

2.2 A Aquisição de Dados em Sistemas de Recomendação

SRs necessitam coletar dados que auxiliem a determinar as preferências dos seus usuários e caracterizar o contexto. A coleta de dados pode ser explícita ou implícita. Na

coleta explícita o usuário entra com os dados (fornecendo uma nota para um conjunto de itens, declara sua localização, etc.). Já na coleta implícita são observadas as ações do usuário (em um site de *e-Commerce*, por exemplo, comprar um item ou adicioná-lo no carrinho de compras indica que o usuário gostou do produto).

Em [Bobadilla et al 2013] é destacado que pode ser observada uma tendência na área de SRs de coletar e integrar diferentes tipos de dados, seguindo a evolução da Web. Neste sentido, os autores destacam que no início da Web (referenciada como *Web 1.0*) os SRs usavam basicamente avaliações dos usuários e seus dados demográficos (informados pelos usuários para a aplicação). Com o advento das redes sociais (*Web 2.0*) SRs passaram a utilizar dados gerados nestas redes e em sites cujo conteúdo é produzido de forma colaborativa. O volume de obtidos de sensores e dispositivos vem crescendo juntamente com a preocupação com a semântica dos dados (*Web 3.0*).

Cabe observar que grande parte do conteúdo disponível na Web ainda é produzido por meio de dados gerados por seres humanos (digitando, pressionando botões, tirando fotos, lendo códigos de barra, etc.). Estas ações exigem atenção e tempo dos usuários, favorecendo a ocorrência de erros [Ashton 2009].

3. Computação Ubíqua

A caracterização do contexto e a coleta de dados que permitem sua caracterização são temas fortemente relacionados à Computação Ubíqua. A Computação Ubíqua é caracterizada pela integração transparente dos recursos computacionais ao dia a dia das pessoas [Weiser 1991]. Além de relacionada à mobilidade (Computação Móvel), a Computação Ubíqua está relacionada à Computação Pervasiva, cuja visão implica em que os dispositivos tenham a capacidade de obter do ambiente dados que permitam criar modelos computacionais para ajustar o comportamento de aplicações [Araújo 2003].

Também relacionada à Computação Ubíqua/Pervasiva está a Internet das Coisas - *Internet of Things - IoT* [Ashton 2009]. A Internet das Coisas enfatiza fato de que objetos físicos (carros, refrigeradores, roupas, etc.) estejam conectados a Internet, possuindo um endereço único, podendo (a partir de sensores) adquirir informações sobre seus estados e sobre o ambiente que os cerca, serem monitorados e comunicar-se entre si sem intervenção humana [Aggarwal et al. 2013].

O cenário proposto pela Internet das Coisas permite vislumbrar uma série de possibilidades em SRs. Geralmente em SRs é apenas possível obter informações sobre uso de objetos digitais (efetuou o *download*, abriu arquivo para leitura, etc.) e não sobre o uso de objetos físicos. Neste sentido, em [Mettouris e Papadopoulos 2014] é observado que um sistema que recomenda livros não tem a possibilidade de obter informações sobre a frequência e por quanto tempo um usuário utiliza um livro. Porém, uma vez que o objeto físico esteja representado na Internet, existe a possibilidade de acompanhar seu uso (o fato de estar fora da estante ou ter sua posição modificada é um forte indicativo do livro estar sendo usado, por exemplo).

4. Sistemas de Recomendação Ubíquos

São considerados Sistemas de Recomendação Ubíquos os SRs que exploram características dos sistemas ubíquos para gerar recomendações. São, portanto sistemas que obtêm vantagens dos avanços da telefonia móvel, das conexões *wireless* e da

capacidade que dispositivos possuem de obter informações sobre o ambiente onde estão [Mettouris e Papadopoulos 2014].

Estes sistemas não representam necessariamente uma nova abordagem para SRs, podendo utilizar abordagens tradicionais (Baseada em Conteúdo, Colaborativa) ou a combinação destas (Híbrida). Cabe ressaltar que Sistemas de Recomendação Ubíquos são sistemas SRs conscientes do contexto. Esta seção apresenta exemplos destes sistemas. É dado ainda destaque para trabalhos desenvolvidos por pesquisadores brasileiros. Alguns dos exemplos citados utilizam poucas informações recuperadas do ambiente (especialmente a localização do usuário).

4.1 Exemplos de Sistemas de Recomendação Ubíquos

Muitos SRs que utilizam recursos da Computação Móvel/Ubíqua estão relacionados ao domínio do turismo. Em [Zheng et al. 2011], por exemplo, é apresentado um SR que recomenda locais e prováveis parceiros de viagem (pessoas que visitaram os mesmos locais e/ou locais similares). No sistema, a similaridade dos usuários é medida pela comparação das sequências dos locais visitados. Já em [Savage et al. 2001] é apresentado um sistema que recomenda lugares de interesse, analisando o perfil social na internet do usuário e os locais que ele frequenta. Este sistema também define as recomendações considerando o humor do seu usuário (informado explicitamente pelo usuário). Por fim, em [Saez-Trumper et al. 2012] é apresentado um SR de eventos que utiliza informações sobre preferências e localização do usuário. Os autores identificaram que as recomendações melhor avaliadas foram aquelas que levaram em consideração a proximidade do usuário do local do evento, fato que ressalta a importância das informações sobre a localização.

Sistemas de Recomendação Ubíquos vêm sendo criados para outros domínios além do turismo. Em [Lin et al. 2011], por exemplo, é apresentado um sistema de recomendação de locais que proporcionam a realização de atividades saudáveis. Este sistema avalia o perfil do usuário, o clima, o tempo, a distância do usuário dos pontos a serem recomendados e também a agenda de atividades do usuário. Já em [Woerndl et al., 2011] é descrito um sistema que considera o combustível disponível, a localização do carro e dos postos de combustível para recomendar o reabastecimento do carro. No sistema apresentado em [Quercia e Capra 2011] é feita a recomendação de amigos em redes sociais a partir da detecção da proximidade física (detectada por meio de telefones celulares - *Bluetooth*). Assim, são identificadas pessoas que frequentam os mesmos locais (ou locais próximos) e o sistema considera esta informação para, dentro de uma rede social, gerar listas de recomendação de pessoas. Já em [Cheng e Shen 2014] é apresentado um sistema que leva em consideração a adequação da música ao perfil do usuário e a sua localização (é considerado que o usuário pode preferir ouvir um tipo de música quando está em uma biblioteca e outro quando está fazendo ginástica). Neste trabalho a localização não é automaticamente detectada, sendo indicada pelo usuário.

Alguns trabalhos vêm explorando e discutindo o uso de tecnologias relacionadas à Internet das Coisas (*Radio-Frequency ID – RFID*, por exemplo) em SRs. Em [Walter et al. 2012], por exemplo, é discutido o uso de sistemas de recomendação em lojas de varejo, considerando o uso de etiquetas *RFID* nos produtos. Uma das possibilidades que surge aqui é o de favorecimento de vendas cruzadas. Em lojas de varejo a venda cruzada

é incrementada colocando próximos produtos relacionados. Já com o uso das etiquetas *RFID* a interação do usuário com os produtos é identificada, sendo assim possível recomendar produtos relacionados por meio de um dispositivo móvel (algo que é feito com frequência hoje em sites de *eCommerce*). Outro exemplo do uso da tecnologia *RFID* e da abordagem colaborativa para auxiliar os visitantes de um museu é apresentada em [Huang et al. 2010] e em [Karimi et al. 2012] onde durante a visita, a interação dos visitantes com os objetos do museu (que possuem *tags RFID*) é acompanhada, sendo feitas recomendações aos usuários. O visitante pode ainda, acessar informações sobre sua visita no museu após finalizá-la e também receber recomendações para exposições futuras. Usando também etiquetas *RFID* nos objetos, em [Yao et al. 2014] é proposta a recomendação de objetos físicos presentes em um ambiente para seus frequentadores - o sistema procura prever o uso de um objeto (um utensílio de cozinha, por exemplo)

4.2 Trabalhos de Pesquisadores Brasileiros

É possível identificar trabalhos de pesquisadores brasileiros que envolvem SRs e recursos relacionados à Computação Ubíqua. Alguns destes trabalhos são apresentados a seguir.

No domínio do turismo, um exemplo está em [Moura et al. 2013] onde é descrito um SR que faz uso de uma ontologia para realizar a recomendação considerando informações do contexto em que o usuário está, tais como: a localização, preferências (alimentação, meios de locomoção, etc.) e restrições (tempo e dinheiro). Outro exemplo é [Marinho et al. 2012], onde as coordenadas geográficas obtidas através de fotos (extraídas do *Panoramio*) e a data em que a foto foi tirada, são usadas para recomendar a usuários lugares de uma cidade a serem visitados em uma determinada época do ano.

Já em [Oliveira et al. 2012] o foco da recomendação é voltado para os possíveis filmes de interesse ao usuário e o local mais próximo em que o usuário irá encontrá-los. O sistema foi pensado para ocasiões específicas como a Copa do Mundo ou as Olímpiadas. Alguns trabalhos estão relacionados a outros domínios de aplicação. O trabalho de [Tito et al. 2013], por exemplo apresenta a proposta de um sistema que considera informações contextuais dos usuários e do trânsito para recomendar rotas de ônibus aos passageiros. Os dados que são levados em consideração para a recomendação são: o deslocamento dos veículos, as características dos passageiros e os fatores dinâmicos que podem afetar o transporte como, por exemplo, a situação climática, entre outros. Já em [Lemos et al. 2012] é apresentado um sistema que recomenda fotos para usuários a partir da similaridade do contexto relacionado as fotos (localização e aspectos temporais são considerados).

Em [Silva et al., 2009] é apresentada uma proposta de arquitetura sensível ao contexto, denominada *PersonalTVware* para suporte a recomendação personalizada de conteúdo para TV Digital. É avaliado nessa proposta quem é o usuário que está assistindo à televisão naquele momento, onde o mesmo está localizado, como está assistindo à televisão (dispositivo móvel, portátil ou fixo com suporte a HDTV, etc.) e quando normalmente um usuário assiste um determinado gênero de programa de TV. Obviamente é levado em conta também o conteúdo que o usuário considera relevante.

Como já mencionado, técnicas de recomendação podem ser utilizadas não apenas para recomendação de itens, mas para determinação de ações. Neste sentido em [Lima et al. 2011], técnicas utilizadas em sistemas de recomendação são empregadas para auxiliar no processo de autenticação dos usuários de dispositivos móveis de forma a evitar que o usuário precise realizar sua autenticação frequentemente (digitando a senha, por exemplo).

É possível identificar uma série de trabalhos de pesquisadores brasileiros que tem como foco a recomendação de objetos de aprendizagem em Ambientes Virtuais de Aprendizagem. Em [Machado e Palazzo de Oliveira 2014], por exemplo, é apresentado o CARLO – *Model for Context-Aware Recommendation of Learning Objects* (Modelo para Recomendação Sensível ao Contexto de Objetos de Aprendizagem) que considera quatro dimensões: perfil do usuário, localização, informações sobre elementos tecnológicos e sobre os objetos de aprendizagem. Os objetos de aprendizagem são então recomendados a partir da adequação destes objetos a instância do modelo, sendo utilizadas regras semânticas e um motor de inferência. Outros exemplos voltados para Ambientes Virtuais de Aprendizagem são [Silva et al. 2013] [Cazella et al. 2014].

5. Perspectivas e Desafios

Uma análise dos trabalhos citados que envolvem SRs e Computação Ubíqua (Tabela 1) permite constatar que a maioria dos trabalhos descritos no presente artigo (10 trabalhos) considera basicamente localização do usuário, informação que hoje é facilmente obtida por meio dos dispositivos móveis. Alguns trabalhos identificados (4 trabalhos) obtêm outros dados do ambiente ou usam (por vezes apenas propõe o uso) recursos relacionados à Internet das Coisas (5 trabalhos). A partir desta análise, é possível constatar que existem muitas possibilidades ainda inexploradas.

Tabela 1. Comparando os trabalhos descritos na seção 4.

Localização do usuário	Outros dados contextuais	Internet das Coisas
[Zheng et al. 2011]	[Lin et al. 2011]	[Walter et al. 2012]
[Savage et al. 2001]	[Tito et al. 2013]	[Huang et al. 2010]
[Saez-Trumper et al. 2012]	[Woerndl et al., 2011]	[Karimi et al. 2012]
[Quercia e Capra 2011]	[Lemos et al. 2012]	[Yao et al. 2014]
[Cheng e Shen 2014]		[Machado e Palazzo de Oliveira 2014]
[Silva et al., 2009]		
[Lima et al. 2011]		
[Moura et al. 2013]		
[Marinho et al. 2012]		
[Oliveira et al. 2012]		

Neste sentido, atualmente celulares incorporam recursos que proporcionam a obtenção de outras informações e permitem verificar, por exemplo, se o usuário está parado, caminhando, correndo, etc. [Lane et al. 2010] [Lara e Labrador 2013] [Mann 1997]. No contexto da *Wearable Computing*, outros dispositivos estão sendo desenvolvidos (óculos, relógios, etc). Informações obtidas por meio destas tecnologias poderão ser úteis na implementação de *Ambient Assisted Living* [Papangelis et al. 2011].

Outro aspecto reside no fato de que uma preocupação nos SRs está em produzir recomendações no momento adequado. Em SRs Ubíquos esta necessidade assume maior importância do que naqueles que levam em conta apenas avaliações históricas. Neste sentido, em um sistema de recomendação de restaurantes, por exemplo, além do gosto do usuário, a sua localização atual e a situação do restaurante (lotado ou não) devem ser levadas em conta. De forma similar, se são recomendados objetos físicos e não apenas objetos digitais é preciso considerar sua disponibilidade, uma vez que pode não ser possível o uso simultâneo de um objeto por diferentes usuários. Este fato é mencionado em [Yao et al., 2014] que ressaltam a necessidade de que objetos físicos possuam um atributo que indique sua disponibilidade e exemplificam isto com a recomendação de utensílios domésticos.

Outra preocupação nos SRs Ubíquos está no volume de dados a ser processado e no tempo para gerar a recomendação. O volume de dados produzido por sensores conectados a Internet irá requerer novas soluções em termos de processamento de dados que envolvem questões de escalabilidade, processamento distribuído e análise em tempo real [Aggarwal et al. 2013]. Neste sentido, em [Rego et al. 2013] o uso do *MapReduce* em um sistema de recomendação é avaliado.

Por fim, cabe ressaltar que questões de privacidade, uma preocupação presente em SRs já há algum tempo [Shokri 2009], assumem ainda maior relevância na Computação Ubíqua, uma vez que um número maior de dados sobre ações dos usuários e sobre objetos estarão disponíveis [Bettini e Riboni 2014] [Langheinrich et al. 2007] [Ziegeldorf et al. 2013].

6. Considerações Finais

O presente trabalho apresentou algumas possibilidades geradas a partir da interação entre a área de SRs e a Computação Ubíqua. Foram apresentados conceitos básicos das duas áreas e exemplos de SRs que podem ser considerados SRs Ubíquos. Finalmente, foram feitas algumas considerações quanto a perspectivas e desafios inerentes ao desenvolvimento de SRs Ubíquos.

Agradecimentos

Igor Eduardo Viana Rudel é bolsista PIBIC/CNPq e foi bolsista do Programa de Bolsas de Iniciação Científica e Iniciação Tecnológica do Colégio Politécnico da UFSM

Referências

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6), 734-749.
- Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1), 103-145.
- Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook* (pp. 217-253). Springer US.

- Aggarwal, C. C., Ashish, N., & Sheth, A. (2013). The internet of things: A survey from the data-centric perspective. In *Managing and mining sensor data* (pp. 383-428). Springer US.
- Araújo, R. B. (2003). Computação ubíqua: Princípios, tecnologias e desafios. In *XXI Simpósio Brasileiro de Redes de Computadores* (Vol. 8, pp. 11-13).
- Ashton, K. (2009). That ‘internet of things’ thing. *RFID Journal*, 22, 97-114.
- Bettini, C., & Riboni, D. (2014). Privacy protection in pervasive systems: State of the art and technical challenges. *Pervasive and Mobile Computing*.
- Bobadilla, J. Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109-132.
- Cazella, S. C., Barbosa, J. L. V., Reategui, E. B., Behar, P. A., & Acosta, O. C. (2014) Recommending Academic Papers for Learning Based on Information Filtering Applied to Mobile Environments. In: Francisco Milton Mendes Neto. (Org.). Technology Platform Innovations and Forthcoming Trends in Ubiquitous Learning. 1ed.: IGI Global
- Chen, A. (2005). Context-aware collaborative filtering system: Predicting the user’s preference in the ubiquitous computing environment. In *Location-and Context-Awareness* (pp. 244-253). Springer Berlin Heidelberg.
- Cheng, Z., & Shen, J. (2014). Just-for-me: An adaptive personalization system for location-aware social music recommendation. In *Proceedings of International Conference on Multimedia Retrieval* (p. 185). ACM.
- Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, 5(1), 4-7.
- Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., & Reiterer, S. (2013). Toward the next generation of recommender systems: applications and research challenges. In *Multimedia Services in Intelligent Environments* (pp. 81-98). Springer International Publishing.
- Herlocker, J. L., & Konstan, J. A. (2001). Content-independent task-focused recommendation. *Internet Computing, IEEE*, 5(6), 40-47.
- Huang, Y. P., Chang, Y. T., & Sandnes, F. E. (2010). Experiences with RFID-based interactive learning in museums. *International Journal of Autonomous and Adaptive Communications Systems*, 3(1), 59-74.
- Karimi, R., Nanopoulos, A., & Schmidt-Thieme, L. (2012). RFID-enhanced museum for interactive experience. In *Multimedia for Cultural Heritage* (pp. 192-205). Springer Berlin Heidelberg.
- Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. T. (2010). A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9), 140-150.
- Langheinrich, M. (2007). RFID and privacy. In *Security, Privacy, and Trust in Modern Data Management* (pp. 433-450). Springer Berlin Heidelberg.

- Lara, O. D., & Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3), 1192-1209.
- Lemos, F. D., Carmo, R. A., Viana, W., & Andrade, R. (2012). Improving photo recommendation with context awareness. In *Proceedings of the 18th Brazilian symposium on Multimedia and the web* (pp. 321-330).
- Lima, J. C. D., Rocha, C. C., Vieira, M. A., Augustin, I., & Dantas, M. A. (2011). Cars-ad: a context-aware recommender system to decide about implicit or explicit authentication in ubihealth. In *Proceedings of the 9th ACM international symposium on Mobility management and wireless access* (pp. 83-92).
- Lin, Y., Jessurun, J., de Vries, B., & Timmermans, H. (2011). Motivate: Towards context-aware recommendation mobile system for healthy living. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th* (pp. 250-253).
- Machado, G. M., & Palazzo de Oliveira, J. P. (2014) CARLO: Modelo Ontológico de Contexto para Recomendação de Objetos de Aprendizagem em Ambientes Pervasivos In SBBD 2014.
- Mann, S. (1997). Wearable computing: A first step toward personal imaging. *Computer*, 30(2), 25-32.
- Marinho, L. B., Nunes, I., Sandholm, T., Nóbrega, C., Araújo, J., & Pires, C. E. S. (2012). Improving location recommendations with temporal pattern extraction. In *Proc. of the 18th Brazilian symposium on Multimedia and the web* (pp. 293-296).
- McDonald, D. W. (2003). Ubiquitous recommendation systems. *Computer*, 36(10), 111-112.
- Mettouris, C., & Papadopoulos, G. A. (2014). Ubiquitous recommender systems. *Computing*, 96(3), 223-257.
- Moura, H., da Costa, C. A., Rigo, S., Silva, E. F., & Barbosa, J. V. (2013). Developing a ubiquitous tourist guide. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web* (pp. 59-66).
- Tito A. O., Ristar, A. R. R., dos Santos, L. M., V Filho, L. A., Tedesco, P. R., & Salgado, A. C. (2013) RecRoute: Uma Proposta de Aplicativo para Recomendação de Rotas de Ônibus Utilizando Informações Contextuais dos Usuários. In SBSI 2013.
- Oliveira, O. C. S., Nunes, M. A. S. N., & Cazella, S. C. (2012). Personal_Movie-Um modelo de Sistema de Recomendação de filmes geolocalizados em eventos. *Revista de Sistemas de Informação da FSMA*, (10), 44-52.
- Papangelis, A., Galatas, G., & Makedon, F. (2011). A recommender system for assistive environments. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments* (p. 6).
- Pazzani, M. J. (1999) A Framework for collaborative, content-based anddemographic filtering. *Artificial Intelligence Review*, Hingham, v.13, n.5, p. 393-408.
- Quercia, D., & Capra, L. (2009). FriendSensing: recommending friends using mobile phones. In *Proceedings of the third ACM conference on Recommender systems* (pp. 273-276). ACM.

- Rego, P. A., Lemos, F. D., Viana, W., Trinta, F., & de Souza, J. N. (2013). MapReduce performance evaluation for knowledge-based recommendation of context-tagged photos. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web* (pp. 249-256). ACM.
- Ricci, F. (2010). Mobile recommender systems. *Information Technology & Tourism*, 12(3), 205-231.
- Saez-Trumper, D., Quercia, D., & Crowcroft, J. (2012, September). Ads and the city: considering geographic distance goes a long way. In *Proceedings of the sixth ACM conference on Recommender systems* (pp. 187-194). ACM.
- Savage, N. S., Baranski, M., Chavez, N. E., & Höllerer, T. (2012). *I'm feeling loco: A location based context aware recommendation system* (pp. 37-54). Springer Berlin Heidelberg.
- Shokri, R., Pedarsani, P., Theodorakopoulos, G., & Hubaux, J. P. (2009). Preserving privacy in collaborative filtering through distributed aggregation of offline profiles. In *Proceedings of the third ACM conference on Recommender systems* (pp. 157-164).
- Silva, F. S., Alves, L. G. P., & Bressan, G. (2009). PersonalTVware: A proposal of architecture to support the context-aware personalized recommendation of TV programs. In *European Interactive TV Conference (EuroITV 2009), Leuven, Belgium*.
- Silva, L. C., Mendes Neto, F. M., & Jácome Júnior, L. (2013). MobiLE: Um Ambiente Multiagente de Aprendizagem Móvel Baseado em Algoritmo Genético para Apoiar a Aprendizagem Ubíqua. *Revista Brasileira de Informática na Educação*, 21(01), 62.
- Walter, F. E., Battiston, S., Yildirim, M., & Schweitzer, F. (2012). Moving recommender systems from on-line commerce to retail stores. *Information Systems and e-Business Management*, 10(3), 367-393.
- Weiser, M. (1991). The computer for the 21st century. *Scientific american*, 265(3), 94-104.
- Woerndl, W., Huebner, J., Bader, R., & Gallego-Vico, D. (2011). A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 273-276). ACM.
- Yao, L., Sheng, Q. Z., Ngu, A. H., Ashman, H., & Li, X. (2014). Exploring recommendations in internet of things. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 855-858). ACM.
- Zheng, Y., Zhang, L., Ma, Z., Xie, X., & Ma, W. Y. (2011). Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, 5(1), 5.
- Ziegeldorf, J. H., Morschon, O. G., & Wehrle, K. (2013). Privacy in the Internet of Things: threats and challenges. *Security and Communication Networks*.

Um Estudo das Abordagens para Correspondência entre Esquemas Através de Amostras

Jacson L. Matte¹, Denio Duarte¹

¹Universidade Federal da Fronteira Sul - UFFS
Campus Chapecó

jacsonmatte@gmail.com, duarte@uffs.edu.br

Abstract. Schema mappings are high-level specifications that describe the relationship between schemas. In real-life applications schema mappings can be quite complex and, then, users must know very well schemas to be matched. However, there exist applications that users know very well the domain but details of the schemas are not known. Thus, tools that can build a target schema based on samples are demanded by users not experts in computer science but experts in the application domain. This paper studies and analyses three approaches to build a schema based on a source schema and a set of samples. We present algorithms and features of each approach as well a comparison between them.

Resumo. Correspondência entre esquemas são especificações de alto nível cujo objetivo é descrever os relacionamentos entre os esquemas. Em aplicações reais, a correspondência é uma tarefa muito complexa que exige conhecimento preciso dos esquemas envolvidos. Porém, em algumas aplicações, os usuários são especialistas no domínio mas desconhecedores da estrutura do banco de dados. Assim, ferramentas que constroem correspondências baseadas em amostras são utilizadas cada vez mais por usuários não experientes em computação mas que pretendem construir seus próprios bancos de dados a partir de um banco de dados existente e amostras de tuplas que formarão o novo banco de dados. Este trabalho tem como objetivo apresentar a análise de três ferramentas para este fim, através do estudo de seus algoritmos e características.

1. Introdução

A correspondência entre esquemas é um importante problema tratado por pesquisas em banco de dados que é aplicada em várias áreas: integração entre aplicações, *data warehouses*, processamento semântico de consultas, entre outros [Rahm and Bernstein 2001]. A tarefa de correspondência pode ser trabalhosa e longa, assim pesquisas nesta área procuram soluções para que a correspondência seja feita de forma automática e resulte em correspondências corretas.

Dados um esquema de origem S e um esquema de destino T . A correspondência entre esquemas é dada por $M = (S, T, \Sigma)$, onde Σ é um conjunto de regras que representam a correspondência dos elementos em S com os elementos em T (também chamados de expressões de mapeamento). Existem basicamente duas técnicas para construir Σ [Rahm and Bernstein 2001, Shvaiko and Euzenat 2005, Qian et al. 2012]: as expressões de correspondência são criadas a partir dos elementos dos esquemas (corres-

pondência baseada em elementos), e as expressões são criadas baseadas nos elementos e nos dados presentes nos esquemas (correspondência baseada em instâncias).

Geralmente, a correspondência entre esquemas é feita com auxílio de usuários conhecedores da aplicação tanto no nível externo quanto no nível lógico. Isso impede que usuários especialistas na aplicação mas sem conhecimento da estrutura do esquema do banco de dados possam construir suas próprias correspondências. Para atender esses tipos de usuários, alguns trabalhos tratam o problema da seguinte forma: dados uma base de dados de origem com o esquema possivelmente desconhecido pelo usuário e um conjunto de tuplas (possivelmente unitário) informado pelo usuário, é gerado um esquema de saída baseado nas duas entradas.

Neste contexto, este trabalho tem como objetivo apresentar um estudo e uma análise de três ferramentas baseadas em amostras (tuplas) e um esquema de entrada para construir correspondência entre esquemas. Os trabalhos selecionados foram: *MWEAVER* [Qian et al. 2012], *FILTER* [Shen et al. 2014] e *DISCOVER* [Hristidis and Papakonstantinou 2002]. Essas ferramentas se apoiam em amostras de usuário para realizar a tarefa de correspondência entre esquemas e, através das amostras e uma base de dados de origem, gerar sua base de dados de destino e, opcionalmente, a consulta correspondente.

Como resultado da análise são apresentadas as técnicas utilizadas por cada ferramenta bem como o escopo de suas aplicações. A metodologia de apresentação segue os seguintes passos: (i) a descrição breve da ferramenta, (ii) a apresentação de um pseudo-código que implementa as técnicas da ferramenta e, (iii) um exemplo do funcionamento da ferramenta. A Figura 1 apresenta um esquema e sua instância de um banco de dados acadêmico simplificado. Esse banco de dados é utilizado como entrada para exemplificar o funcionamento das ferramentas estudadas.

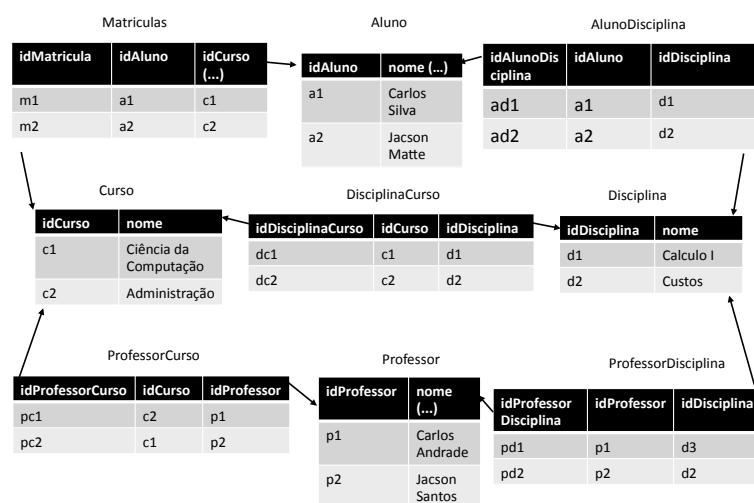


Figura 1. Modelo lógico simplificado e instância de um sistema acadêmico.

O restante do trabalho está organizado da seguinte forma: a próxima seção apresenta abordagens de correspondência entre esquemas por amostras estudadas, a Seção 3 apresenta o resultado da análise e a Seção 4 apresenta a conclusão deste artigo.

2. Abordagens Estudadas para Correspondência entre Esquemas

Esta seção apresenta as ferramentas estudadas de correspondência entre esquemas a partir das entradas de usuário (amostras).

2.1. MWEAVER

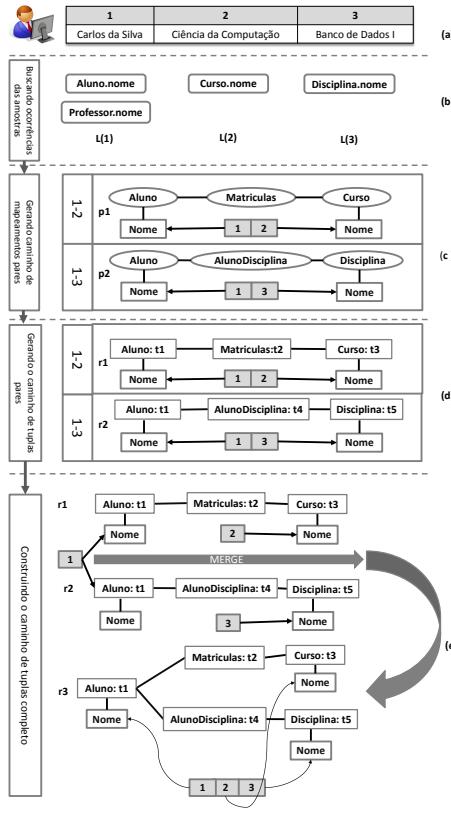


Figura 2. Exemplo do algoritmo TPW.

MWEAVER é uma ferramenta proposta por [Qian et al. 2012] que constrói, automaticamente, o mapeamento entre esquemas a partir de amostras fornecidas pelo usuário. A ferramenta é composta pelo algoritmo de correspondência entre esquemas chamado *TPW (tuple path weaving)*. Seu funcionamento está dividido em cinco etapas principais: buscar ocorrências das amostras; gerar caminho de mapeamentos pares; gerar o caminho de tuplas pares; construir o caminho de tuplas completo e, classificar os mapeamentos. A Figura 2 apresenta as quatro primeiras etapas do processo. O Algoritmo 1 representa todas as etapas para geração dos mapeamentos candidatos.

Algoritmo 1: *TPW*

```
1 Entrada : Base de dados D, amostras de usuário ( $E_1, \dots, E_n$ ) e  
grafo G do esquema ;  
2 Saída : Mapeamento candidato M;  
3 MapaLocalizacao L, MapeamentoCandidato M;  
4 L  $\leftarrow$  CriaMapaLocalizacao(D,  $E_s$ );  
5 CamMapPar  $\leftarrow$  CriaCamMapPar(L, G);  
6 foreach CMP in CamMapPar do  
7   | if SQL(CMP) != "empty" then  
8   |   | CamTuplePar += CMP;  
9   | end  
10 end  
11 M  $\leftarrow$  CamTuplaPar;  
12 foreach CTP in CamTuplePar do  
13   | if !weave(M, CTP + 1) then  
14   |   | return -1;  
15   | end  
16 end  
17 ranking(M);
```

2.1.1. Buscar ocorrências das amostras

Na primeira etapa do Algoritmo 1, busca-se os atributos no banco de dados de origem que contém pelo menos uma amostra, as entradas para essa etapa realizada pela função *CriaMapaLocalização* (linha 4) são a base de dados D e as entradas de usuário E_s que retorna o mapa de localização L . Por exemplo, na Figura 2 que considera como entrada o banco de dados da Figura 1, o usuário digitou os seguintes dados de entrada: *Carlos da Silva, Ciência da Computação e Banco de Dados I* (Figura 2 (a)). Após isso, procura-se primeiro pela amostra *Carlos da Silva*, no banco de dados de origem D , o resultado seria $L(1) = \{Aluno.nome, Professor.nome\}$, pois esses são os atributos que contém *Carlos da Silva*. Da mesma forma são criados todos os L_s , como mostrado na Figura 2 (b). L é chamado de *mapa de localização*.

2.1.2. Geração de caminho de mapeamentos pares

Os caminhos de mapeamento são gerados pela função *CriaCamMapPar* (Linha 5) que recebe como parâmetro o *mapa de localização* L e o grafo dirigido G (para o estudo de caso, G representa o modelo lógico da Figura 1), onde os nodos representam as tabelas e os vértices as relações de chave-estrangeira. Os caminhos de mapeamento pares representam o caminho entre os itens do mapa de localização L no grafo G .

Por exemplo, dado dois elementos extraídos do mapa de localização L : *Aluno.nome* e *Curso.nome*. Um caminho de mapeamento são as tabelas que contém os elementos e possuem uma ligação no grafo do esquema G . Neste exemplo, esse caminho é entre as tabelas *Aluno* e *Curso*, através da tabela *Matriculas*. O caminho no grafo também é delimitado pela quantidade de junções entre as tabelas. Duas junções são o suficiente para representar o relacionamento da tabela *Aluno* e *Curso*. Caso não encontre

um caminho no grafo G ou excede a quantidade de junções, o caminho será considerado inválido. O número da coluna na tabela de entrada indica o valor das chaves; as chaves 1 e 2 do primeiro caminho de mapeamento gerado, indicam os elementos extraídos de $L(1)$ e $L(2)$ na Figura 2 (b); (as setas nas etapas (c) e (d) da Figura 2 indicam a relação das chaves com os atributos).

2.1.3. Construção do caminho de tuplas completo

Nesta etapa, inicialmente, é gerado o caminho de tuplas pares. Na linha 8 do Algoritmo 1, são adicionados a um caminho de tupla todos os caminhos de mapeamento que sejam válidos. Para isso, é gerado uma consulta *SQL* em cada caminho de mapeamento, se o retorno não é vazio, o caminho de mapeamento é válido, senão é inválido. Assim, cada caminho de mapeamento válido resulta em um caminho de tupla.

Em seguida, é realizado a fusão entre esses caminhos (linhas 11 a 16 do Algoritmo 1). A partir de um mapeamento candidato M inicial (linha 11) é realizado a fusão com os caminhos de tuplas restantes. A fusão termina quando não é mais possível fundir dois vértices, retornando um valor negativo (linha 14). Por fim, além do mapeamento r_3 da Figura 2 (e), outro mapeamento candidato seria gerado apenas substituindo a tabela *Professor* pela *Aluno*.

O Algoritmo 1 faz a busca na base de dados por texto completo. Além disso, uma pontuação é mostrada a cada mapeamento candidato, formando um *ranking* entre os mapeamentos candidatos (linha 17), sendo a média de duas notas: a primeira nota indica a qualidade das amostras, coincidindo com os dados reais das tuplas e, a segunda é uma pontuação de complexidade, sendo o número de associações na correspondência.

2.2. DISCOVER

DISCOVER [Hristidis and Papakonstantinou 2002] é uma ferramenta que resulta em redes de consultas candidatas oriundas de um plano gerador. Essas redes são criadas por um conjunto de tuplas fornecidas pelo usuário, estando associadas pela relação de chave-primária e estrangeira. *DISCOVER* realiza duas etapas para gerar as redes de consultas candidatas: (i) o plano gerador de redes candidatas gera todas as redes de consultas candidatas e, (ii) faz uma avaliação sobre as redes candidatas, a fim de reutilizar expressões em comum entre elas.

O Algoritmo 2 apresenta a implementação da construção das redes de consultas candidatas. Tendo como entrada um grafo orientado G representando o esquema da base de dados da Figura 1, um tamanho T que limita o número de junções em uma rede e, o conjunto de palavras-chave K fornecidas pelo usuário. Já como saída, é apresentado o conjunto de redes candidatas que contém todas as palavras-chave sem redundância. Na linha 3, a estrutura CT_s contém todas as tuplas do banco que dados que continham as palavras-chaves. Na linha seguinte, é selecionada uma palavra-chave inicial de forma aleatória pela função *selecionaChave* que tem como parâmetro todas as palavras-chave. Após selecionada a palavra-chave, são geradas as redes de consultas candidatas (linha 6), através de uma busca no grafo G das palavras-chave e, armazenado o conjunto de redes candidatas na variável QC_s , realizada pela função *geraRedesCandidatas* que tem como parâmetro uma tupla C e o conjunto de palavras-chave K .

Algoritmo 2: DISCOVER

```

1 Entrada : Grafo de esquema  $G$ , Tamanho  $T$ , palavras-chave  $K$ ;
2 Saída : Rede de consultas candidatas  $QC_s$ ;
3 conjuntoTuplas  $CT_s$ ;
4  $k = \text{selecionaChave}(K)$ ;
5 foreach  $C$  in  $CT_s$  do
6   |  $QC_s = \text{geraRedesCandidatas}(C, k)$ ;
7 end
8 foreach  $C$  in  $QC_s$  do
9   | if  $C$  succeeds pruning then
10    |   |  $QC_s \leftarrow QC_s - C$ ;
11    |   | else
12    |   |   | continue;
13    |   | end
14 end
15 end
16 return  $(QC_s)$ 

```

Após, são avaliadas as condições de eliminações dessas redes (linhas 9-14). Primeiro são eliminadas as redes candidatas que possuam relações repetidas com a mesma palavra-chave e, depois as redes em que suas folhas sejam relações que não possuam palavras-chave. Dados G e K (conjunto de tuplas da Tabela 1), a construção das redes é feita em duas etapas principais: a geração do conjunto de todas as redes candidatas e a avaliação das redes candidatas.

	A	B	C
1	Carlos	Computação	
2	Jacson		Custos

Tabela 1. Amostras

A geração do conjunto de redes candidatas inicia lendo a primeira tupla de K , ou seja, *Carlos* e *Computação*. Assim, uma busca no grafo G é feita para encontrar um caminho entre essas palavras-chave. Para que o caminho ou a rede sejam válidos, eles devem conter todas as palavras-chave, a menos que existam tabelas repetidas como exemplo o caminho: $\text{Aluno}_{\langle\text{Carlos}\rangle} \bowtie \text{Matricula} \bowtie \text{Aluno}_{\langle\text{Carlos}\rangle}$. Um caminho como $\text{Aluno}_{\langle\text{Carlos}\rangle} \bowtie \text{Matricula} \bowtie \text{Curso}_{\langle\text{Computacao}\rangle}$ é considerado válido. São consideradas redes inválidas aquelas que não atenderem as condições de eliminação como: $\text{Aluno}_{\langle\text{Carlos}\rangle} \bowtie \text{Matricula} \bowtie \text{Curso}_{\langle\text{Computacao}\rangle} \bowtie \text{DisciplinaCurso}$ (possui folha sem palavra-chave).

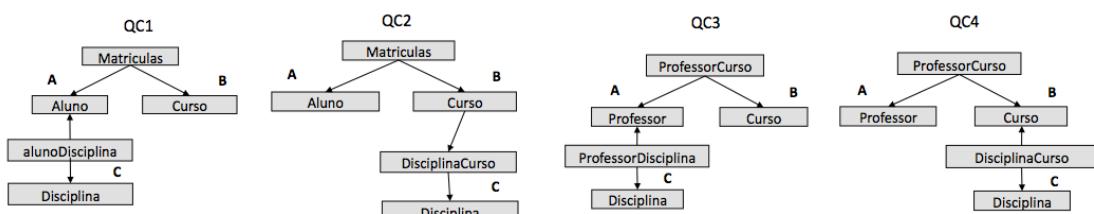


Figura 3. Consultas candidatas.

Na segunda etapa, *DISCOVER* avalia as redes candidatas para verificar quais são as redes inválidas e as redes candidatas que compartilham expressões de junções para construir um conjunto de expressões intermediárias e usá-las em outras redes candidatas. O plano gerador contém um plano de execução, que calcula e utiliza os resultados intermediários na avaliação das redes candidatas. Por fim, uma instrução *SQL* é produzida para cada tupla do plano de execução e estas instruções são passadas para o *SGBD*, retornando as redes que são as soluções para o problema. Para este exemplo, utilizando a tabela de entrada da Figura 1, o resultado são as redes candidatas $QC1$, $QC2$, $QC3$ e $QC4$ da Figura 3, pois atendem a todas as restrições.

Por fim, é feito um cálculo para o custo do plano de execução, definido pela seguinte fórmula: $\frac{frequencia^A}{log^B(tamanho)}$. Sendo a variável *frequencia* o número de ocorrências de *joins* em uma rede candidata e a variável *tamanho* é o número estimado de palavras-chave na rede candidata e, A e B são constantes pré-definidas. O termo *frequencia^A* indica a quantidade máxima de resultados intermediários reutilizados e o termo $log^B(tamanho)$ indica o tamanho dos resultados intermediários que são calculados primeiro. A variável A está relacionada ao tamanho das redes candidatas e, B a capacidade de reutilização, tendo assumido os valores para $\{A,B\}=\{1,0\}$.

Uma extensão do *DISCOVER* [Hristidis et al. 2003] foi proposta pelos mesmos autores que apresenta uma nova forma de classificar os melhores candidatos, através de técnicas de recuperação de informação ao invés do número de junções.

2.3. FILTER

FILTER [Shen et al. 2014] é uma abordagem de correspondência entre esquemas também baseada em amostras e tem como objetivo descobrir o mínimo de consultas geradas. A correspondência é realizada através de alguns exemplos fornecidos pelo usuário, que podem representar a saída de uma consulta.

Algoritmo 3: FILTER

```

1 Entrada : Filtros  $F_s$  e consultas candidatas  $QC_s$ ;
2 Saída : consultas candidatas válidas  $Q_s$ ;
3  $QX_s \leftarrow QC_s$ ,  $FX_s \leftarrow F_s$ ;
4 while  $QX_s \neq 0$  do
5    $F_i = \text{selecionaProxFiltro}(FX_s)$ ;
6    $F_i = \text{avaliaFiltro}(F_i)$ ,  $FX_s \leftarrow FX_s - F_i$ ;
7   if  $F_i$  succeeds then
8      $F_v \leftarrow F_v + F_i$  ;
9      $Q_i = \text{verificaConsulta}(QX_s, F_v)$ ,  $QX_s \leftarrow QX_s - Q_i$ ;
10     $Q_s \leftarrow Q_s + Q_i$ ;
11   else
12      $F_f \leftarrow F_f + F_i$ ;
13      $Q_i = \text{verificaConsulta}(QX_s, F_f)$ ,  $QX_s \leftarrow QX_s - Q_i$ ;
14   end
15 end
16 end
17 return ( $Q_s$ )

```

O algoritmo de geração das consultas candidatas em *FILTER* é o mesmo proposto por [Hristidis and Papakonstantinou 2002], com uma pequena adaptação que rotula

cada nodo da consulta em uma estrutura de rede como apresentado na Figura 3, através do nome da coluna pertencente à tabela de entrada, que contém a amostra. Por exemplo, na consulta candidata QC_1 , a relação *Aluno* é rotulada com a coluna A .

O Algoritmo 3 avalia as consultas candidatas. Tendo como entrada os filtros F_s que são subestruturas de uma consulta, e as consultas candidatas QC_s , gerando como saída as consultas válidas Q_s . No Algoritmo 3 para cada uma das consultas candidatas, primeiramente é selecionado um filtro por um modelo estatístico (linha 5), sendo realizado pela função *selecionaProxFiltro* que tem como parâmetro o conjunto de filtros FX_s . Logo depois é feita a avaliação do filtro escolhido (linha 6) gerando uma consulta *SQL* para cada linha da tabela. Na mesma linha, o filtro selecionado é excluído da lista dos FX_s .

Após a avaliação, é verificado se o filtro é válido (linha 7), ou seja, verifica se o valor retornado são os mesmos contidos na tabela de entrada. Desta forma, o filtro é adicionado ao conjunto de filtros válidos F_v (linha 8) e, no passo seguinte, a função *verificaConsulta* retorna as consultas não avaliadas QX_s que atendam o filtro F_v (linha 9). Assim, a consulta avaliada é excluída da lista de consultas QX_s e adicionada a lista de consultas válidas Q_s . Se o filtro for inválido, será adicionado ao conjunto de filtro inválidos F_f e testado com cada uma das consultas candidatas não avaliadas (linha 13). Por fim, o algoritmo termina quando todas as consultas foram avaliadas e retorna todas as consultas válidas Q_s .

A Figura 4 mostra três filtros das consultas candidatas QC_3 e QC_4 da Figura 3. Na avaliação de um filtro, um filtro é válido se a linha específica da tabela de entrada é dada como sendo condizente com o resultado de uma consulta *SQL* que é gerada para cada filtro. Esse filtro pode ser verificado utilizando a seguinte sintaxe:

```
SELECT * TOP 1 FROM < Tabela > WHERE < predicado > CONTAINS (< amostra >)
```

Usando o modelo de consulta *SQL*, pode-se verificar o filtro J_1 para primeira linha da tabela de entrada (Tabela 1). O resultado da consulta não é vazio, portanto o filtro J_1 é considerado válido.

Após definir os filtros é necessário saber como utilizá-los. Por exemplo, considere a tabela na Figura 4, através da base de dados na Figura 1 e utilizando as consultas candidatas QC_3 e QC_4 da Figura 3. A Figura 4 apresenta três filtros J_1 , J_2 e J_3 destes candidatos, com as colunas da tabela de entrada projetadas nas relações sublinhadas das consultas candidatas. As colunas A , B e C da tabela do exemplo são mapeadas para as colunas dos filtros. Os pares (J_i, j) são utilizados para caracterizar um filtro para a sub-rede de junção J_i na linha j^{th} do exemplo, na coluna projetada na tabela.

Na Figura 4 são mostrados seis avaliações dos filtros, ou seja, o produto de três filtros com duas linhas da tabela de entrada da Tabela 1. Cada avaliação dos filtros $F = (J_i, j)$ está ligada a um candidato QC . Sendo que os círculos abaixo das avaliações dos filtros indicam: filtro válido quando é preenchido, e não preenchido é inválido. O preenchimento não é conhecido antes de ser avaliado. Os dois candidatos QC_3 e QC_4 são todos inválidos. Para verificar isso é preciso apenas avaliar $(J_1, 1)$. É avaliado se o filtro está contido na consulta candidata, sendo transformado em uma consulta *SQL* e, verificado se o seu resultado não corresponde com as entradas da tabela. Essa avaliação dos filtros com sucesso ou falha se trata de um problema combinatório. Por isso, segundo [Shen et al. 2014], o custo de avaliação é difícil de ser estimado de forma geral.

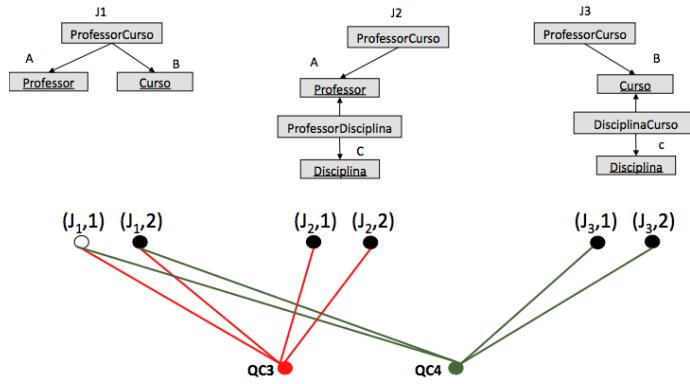


Figura 4. Utilizando filtros para verificação de candidatos.

Para avaliar o custo, foi proposto um modelo probabilístico usado para saber qual o próximo filtro a ser avaliado, ou seja, o que tem mais probabilidade de falha. O custo é definido pela seguinte fórmula: $p(F) = \bar{p} \cdot \frac{n_f}{|col(T)|}$, na qual $p(F)$ é a probabilidade da falha, \bar{p} é uma constante que assume o valor da probabilidade média de falha, n_f é o número de células não vazias em uma linha da tabela, que é mapeada para as colunas de um filtro, e $col(T)$ é o número total de colunas na tabela de entrada.

3. Análise Comparativa

A Tabela 2 apresenta um comparativo das abordagens apresentadas neste artigo para correspondência entre esquemas através de amostras. As seguintes características são consideradas na comparação: (i) a forma de classificar os resultados exibidos pelas ferramentas (coluna *Classificação*); (ii) a estrutura principal utilizada para execução dos algoritmos (coluna *Estrutura*); (iii) se utiliza como técnica de pesquisa no banco de dados aproximada ou por texto completo (coluna *Busca de ocorrências*); (iv) o número mínimo de entradas fornecidas pelo usuário para execução da ferramenta (coluna *Entradas mínimas*); e (v) quem gerencia a memória da ferramenta (coluna *Memória*).

Método	Classificação	Estrutura	Busca de ocorrências	Entradas mínimas	Memória
MWEAVER	Pontuação	Grafo dirigido	Texto completo	Prim. linha completa	SO
DISCOVER	Nº Junções	Grafo dirigido	aproximado	Única amostra	SGBD
FILTER	Probabilístico	Grafo dirigido	aproximado	Única amostra	SGBD

Tabela 2. Comparativo das Abordagens Analisadas.

A forma de classificação utilizada para exibir os resultados para o usuário na ferramenta *MWEAVER* é através de uma pontuação empregada a cada mapeamento candidato. Essa pontuação faz um *ranking* de 0 até 1, apresentando os resultados em ordem decrescente, assim os resultados mais próximos de 1 são considerados mais corretos. Já na ferramenta *DISCOVER* os resultados são classificados pelo número de junções, sendo que os resultados com o menor número de junções são exibidos primeiro para o usuário. *FILTER* não classifica seus resultados, pelo fato dos filtros válidos e inválidos não serem conhecidos e, selecionados por um modelo estatístico afetando uma possível ordem das consultas candidatas.

Uma das características utilizadas pelos algoritmos para a redução do acesso ao banco de dados relacional e das operações de junções é a utilização da estrutura de um

grafo dirigido. Neste grafo, as arestas indicam as relações de chave-estrangeira e os nós das tabelas, pois a busca no grafo que está em memória, torna-se muito mais rápida computacionalmente e, diminui a quantidade de acessos ao SGBDR.

Outra característica importante é a forma de armazenamento dos resultados intermediários obtidos pelos algoritmos. *MWEAVER* delega a gestão de memória dos seus resultados intermediários para a memória principal do sistema operacional. Já *DISCOVER* e *FILTER* armazena os seus resultados intermediários em tabelas temporárias no banco de dados, delegando assim a gestão de memória para o *SGDB*. Desta forma, *MWEAVER* perde em desempenho devido uma grande quantidade de memória utilizada e a troca de dados entre o disco e a memória.

Por fim, as ferramentas se diferenciam quando a forma de busca das ocorrências de amostras na base de dados. *MWEAVER* usa uma busca pelo texto completo, o usuário precisa informar o valor exato que quer encontrar na base de dados. Entretanto, as outras duas abordagens relacionadas na Tabela 2 fazem a busca pelo texto aproximado, o usuário sabendo parte do valor já é o suficiente para fazer a busca.

4. Conclusão

Neste artigo, foi abordado o tema de mapeamento entre esquemas orientado a amostras para facilitar as tarefas de integração de dados para os usuários especialistas. Embora, seja difícil para os utilizadores especialistas entender a semântica precisa de esquemas e mapeamentos, uma maneira de auxiliar é apenas fornecendo exemplos de amostra, facilitando a tarefa.

A análise e o comparativo apresentados neste trabalho tiveram como objetivo apresentar algumas técnicas existentes para a geração de esquemas baseada em amostras. Como aplicação futura deste trabalho, pretende-se propor uma ferramenta que estenda técnicas aqui apresentadas com outras características visando otimizar o resultado (*e.g.*, utilizar os metadados do banco de dados para auxiliar na construção do esquema destino).

Referências

- Hristidis, V., Gravano, L., and Papakonstantinou, Y. (2003). Efficient IR-style keyword search over relational databases. In *Proceedings of the 29th VLDB Endowment*.
- Hristidis, V. and Papakonstantinou, Y. (2002). Discover: Keyword search in relational databases. In *Proceedings of the 28th International Conference on VLDB*, VLDB '02.
- Qian, L., Cafarella, M. J., and Jagadish, H. V. (2012). Sample-driven schema mapping. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*.
- Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350.
- Shen, Y., Chakrabarti, K., Chaudhuri, S., Ding, B., and Novik, L. (2014). Discovering queries based on example tuples. In *Proceedings of the 2014 ACM SIGMOD*, SIGMOD '14.
- Shvaiko, P. and Euzenat, J. (2005). In *Journal on Data Semantics IV*, volume 3730, pages 146–171.

Aplicando Técnicas de Business Intelligence sobre dados de desempenho Acadêmico: Um estudo de caso

Ana Magela Rodriguez Almeida¹, Sandro da Silva Camargo¹

¹ Curso Engenharia de Computação– Universidade Federal do Pampa (UNIPAMPA)
Caixa Postal 07 – 96.400-970 – Bagé – RS – Brasil

{anamagelaa, camargo.sandro}@gmail.com

Abstract. *In the present days, the volumes of data generated and stored by organizations are growing disproportionately. This situation comes along with big challenges, like the integration and transformation of this data in relevant information to optimize the process of decision-making. So it's necessary to build a knowledge base to integrate and organize the available data in order to make them easy to understand and allow the quick access, enabling exploitation and analysis of data in order to meet the strategic expectations of organizations. In this scenario, this paper presenting an approach for implement Business Intelligence in the UNIPAMPA, aiming to significantly improve the resource management application in the organization.*

Resumo. *Atualmente, os volumes de dados gerados e armazenados pelas organizações estão crescendo de forma desmesurada. Esta situação, vem acompanhada de grandes desafios, entre eles, a integração e transformação destes dados em informações relevantes para aprimoramento do processo decisório. Desta forma, faz-se necessária a execução de um complexo processo para organizar e integrar os dados disponíveis de forma facilmente entendida e que permita o acesso rápido, possibilitando a exploração e análise de dados a fim de atender as expectativas estratégicas das organizações. Neste cenário, este trabalho apresenta uma abordagem para implementação de Business Intelligence dentro de uma instituição federal de ensino superior, a fim de otimizar a aplicação de seus recursos.*

1. Introdução

O sucesso de uma organização é dependente de sua capacidade tomar decisões corretas. Assim, o processo de tomada de decisão é um aspecto chave dentro da gestão organizacional. A existência de dados e informações que descrevam o contexto do problema auxilia a maximizar o sucesso das decisões tomadas. Paralelamente, ao longo de sua existência, as organizações geram e armazenam uma grande quantidade de dados referentes as atividades realizadas. Desta forma, a aplicação de técnicas de business intelligence (BI) no processo de tomada de decisão estratégica torna-se de extrema importância, uma vez que decisões equivocadas podem comprometer o futuro de uma organização.

Inseridas no mesmo contexto das demais organizações, instituições de Ensino também necessitam ter recursos para subsidiarem a otimização de seus processos decisórios. Apesar de ter apenas 8 anos, a Universidade Federal do Pampa (UNIPAMPA) já armazena em seu banco de dados mais de um milhão de registros que contam sua história, apontando seus problemas e virtudes. Porém, é necessário

transformar estes dados armazenados em informações úteis para a Instituição. Desta forma, a implementação de soluções BI aplicadas em organizações de ensino podem ser de alta relevância para identificar, compreender e prever possíveis problemas residentes na Instituição, que muitas vezes são resultado de decisões tomadas sem base no conhecimento escondido nestes dados. Desta maneira, a consolidação e exploração destes dados apoiam o processo de tomada de decisões e assim mostram novas estratégias a fim de otimizar a utilização dos recursos da Instituição.

Com base na complexidade em gerenciar e processar grande volume de dados, este trabalho está focado na aplicabilidade de técnicas no contexto BI sobre os dados de desempenho acadêmico de uma instituição de ensino superior. Desta forma, este trabalho visa descrever uma metodologia de aplicação de técnicas de BI com o objetivo de fornecer subsídios para o trabalho dos gestores educacionais, contribuindo para o aprimoramento do processo decisório em uma instituição de ensino superior.

O artigo está estruturado da seguinte forma, na seção 2 é apresentada uma introdução sobre conceitos relevantes a BI, na seção seguinte é descrito o processo de implementação de BI, juntamente com as etapas de criação de *Data Marts* (DM). Na seção 4 é descrito o estudo de caso realizado a partir da plataforma selecionada, assim como uma discussão sobre os resultados.

2. Conceito de Business Intelligence

Business Intelligence consiste em um conjunto de tecnologias, técnicas, conceitos e ferramentas orientadas para análise e apresentação de informações para auxiliar os gestores no processo decisório e como isto permitir às organizações otimizar seus recursos de negócio e alcançar melhores resultados [Wu et. al. 2007].

O principal objetivo de BI é oferecer acesso aos dados de forma simples e, assim proporcionar aos gestores a capacidade de realizar análises convenientes [Turban et. al., 2009]. Ainda para o autor, o processo de BI fundamenta-se na transformação de dados em informações, depois em decisões e, finalmente, em ações. Assim, o propósito de BI é converter volumes de dados em informações novas e úteis, que transformadas em conhecimento podem beneficiar às atividades de uma organização. Para realizar o processo de transformação dos dados em informações, é necessário um ambiente que consolide e permita o uso estratégico dos dados extraídos das bases de dados. Desta maneira, estes ambientes armazenam os dados possibilitando o seu processamento por ferramentas especiais.

2.1. Sistemas de BI

No mercado globalizado existente nos dias de hoje, as organizações cada vez mais buscam destacarem-se e tornarem-se mais competitivas no mercado. Em consequência disto, conseguem otimizar seu desempenho e obterem melhores resultados.

Neste cenário, sistemas de BI proporcionam um ambiente para a unificação dos dados e execução do processo de descoberta de conhecimento. São capazes de extrair, armazenar, processar e interpretar os dados, muitas vezes em tempo real. Estes sistemas podem ser definidos como ferramentas através das quais é possível descobrir conhecimento sobre o histórico operacional da organização, afim de proporcionar subsídios para uma tomada de decisão mais efetiva pelos gestores e, com isso, tornar este processo mais preciso e confiável.

Segundo [Turban et. al. 2009] e [Cano 2007] Sistemas de BI possuem quatro grandes componentes:

- Fontes de Informação: São sistemas dos quais são obtidas as informações, tais como, sistemas operacionais e transacionais da organização.
- Processo de extração, transformação e carga (ETL): Recupera e transforma os dados que serão carregados para a base que consolida os dados para análise.
- *Data Warehouse* (DW): Repositório de dados integrados e não volátil que proporciona informação preparada no processo ETL para a análise.
- *Data Mart* (DM): É uma base de dados específica para uma determinada área dentro de uma organização.
- Área de apresentação: Conjunto de ferramentas de BI que permitem a exploração e visualização da informação armazenada no DW [Kimball e Ross 2002].

2.2. Data Mart

Um *Data Mart* pode ser definido como uma base de dados departamental específica para uma determinada área dentro de uma organização enfatizando o fácil acesso a uma informação relevante. Para [Kimball e Ross 2002] *Data Mart* são subconjuntos de um *Data Warehouse* completo, possuindo as mesmas características.

Para [Diaz e Caralt 2012], o objetivo de construir um DM é responder a uma determinada análise dentro da organização. Geralmente estes sistemas armazenam menos informação que os DW e permitem acesso rápido das informações para análise, pois possuem indexação de armazenamento. Podem ser tanto dependentes como independentes do DW, constituídos pelas arquiteturas *Top-Down* e *Bottom-Up*.

2.3. Modelagem de Dados

A modelagem é um sistema para concepção e visualização de um modelo de dados. Para a modelagem dos dados armazenados em um *Data warehouse* e em *Data Mart* é utilizada a modelagem dimensional. Para [Kimball e Ross 2002] modelagem dimensional é uma técnica que possibilita a criação de um modelo de dados dimensional.

Este modelo dimensional é constituído por um conjunto de medidas que descrevem aspectos de negócios. Esta modelagem permite sumarizar e estruturar os dados para dar suporte à análise de dados. Três elementos formam este modelo, são eles: Fatos, Dimensões, Medidas.

De acordo com [Kimbal e Ross 2002], a tabela de fatos é a principal tabela do modelo. Os fatos são coleções de itens de dados, compostas de dados de medida e de contexto. Estas coleções são compostas pelas medições numéricas que representam a evolução dos negócios de uma organização. O fato registra o dado que será analisado e é composto pela chave primária e um conjunto único de valores de chaves de dimensões.

As tabelas de dimensão contém as descrições de negócio, são os elementos dos fatos do negócio. Cada dimensão pode ter vários níveis hierárquicos para proporcionar um melhor entendimento e uma melhor visualização dos indicadores. Os atributos das dimensões são os principais atributos usados para obter vistas do processo de negócio,

tais como filtro nas consultas, agrupamentos e relatórios [Diaz e Caralt 2012]. As medidas ou variáveis são os atributos numéricos que representam os indicadores que mostram a evolução do negócio da empresa [Turban et. al. 2009].

Existem principalmente duas abordagens dentro da modelagem multidimensional, o modelo *Star Schema* e o modelo *Snow Flake* [Diaz e Caralt 2012]. O *Star Schema* consiste em estruturar informação em processos, vistas e medidas em forma de estrela [Diaz e Caralt 2012]. Tem como característica básica a presença de dados redundantes para proporcionar um melhor desempenho [Junior 2004]. Em termos de desenho, este esquema é composto por uma tabela de fatos no centro para o fato objeto de análise, e uma ou várias tabelas auxiliares chamadas tabelas de dimensões para cada ponto de vista da análise que participa da descrição do fato [Diaz e Caralt 2012]. Neste esquema, a consulta ocorre inicialmente nas tabelas de dimensões e depois na tabela de fatos, garantindo a precisão dos dados através de uma estrutura completa [Junior 2004].

O segundo modelo, o *Snow Flake* é um esquema derivado do modelo *Star Schema*, onde as tabelas de dimensões se normalizam em diversas tabelas. Com isso a tabela de fatos deixa de ser a única tabela que se relaciona com as outras, e assim, surgem novas uniões [Diaz e Caralt 2012].

2.4. Processo de Extração, Transformação e Carga - ETL

O processo ETL é de extrema importância na construção de um sistema BI, pois comprehende os procedimentos realizados em torno do DW para coleta e transformação dos dados antes de serem carregados no armazém de dados [Diaz e Caralt 2012]. Este processo consiste na utilização de ferramentas que realizarão estas etapas de coleta, limpeza e migração dos dados ao DW. Segundo [Junior 2004], este processo apresenta-se em três camadas: Extração, Transformação e Carga.

2.5. Ferramentas OLAP

Ferramentas OLAP consistem em um conjunto de técnicas voltadas para acesso e análise *ad-hoc* de dados, utilizando uma série de recursos para exploração destas informações. Estas aplicações baseadas em *On-Line Analytical Processing*, OLAP, referem-se ao conjunto de processos para integração, análise e manipulação de grande volumes de dados, disponibilizando uma série de funcionalidades, objetivando uma maior compreensão destes dados por analistas e gestores no processo de análise corporativa.

Os autores [Diaz e Caralt 2012] explicam o conceito como uma tecnologia que permite instanciar os dados em uma visão multidimensional, permitindo a apresentação das informações em distintas perspectivas. A multidimensionalidade é um conceito chave de uma ferramenta OLAP para sintetizar informações, refere-se à visão conceitual personalizada da informação alvo de análise, ou seja, é possível obter diferentes análises a partir da mesma base de dados possibilitada pela mudança entre as diferentes perspectivas [Cano 2007].

2.6. SpagoBI

É uma suíte completa que oferece suporte a negócios cotidianos e estratégicos, tanto em nível de tomada de decisões quanto em nível operacional. Suporta ainda, *Data Mining*,

permitindo descobrir padrões, processo ETL para carregar e gerir a *Data Warehouse*, OLAP que permite a análise multidimensional dos dados, entre outros.

O modelo analítico está formado por diversos motores analíticos, entre os mais usados estão os Relatórios, Indicadores *Key Performance Indicator* (KPI), Painel de Controle, Gráficos, Análise geográfico e Análise multidimensional de dados OLAP.

Assim, esta ferramenta oferece uma solução de BI flexível que proporciona suporte ao monitoramento, análise e apresentação de dados. Desta forma, toda a escalabilidade necessária para a organização é suportada pelo SpagoBI, em termos de arquitetura, funcionalidades e segurança.

3. Implementação de BI

A estratégia utilizada para o desenvolvimento do DM envolve as principais etapas do modelo proposto por [Kimball e Ross 2002], que iniciou-se com o planejamento do projeto e a elaboração do entendimento das necessidades para implementação de BI na UNIPAMPA, assim como, das técnicas e ferramentas necessárias para o desenvolvimento do trabalho.

Um requisito primordial para a criação de DM é a forma com que os dados são organizados nas bases de dados, para que as consultas sejam relevantes no processo decisório. Isto se torna possível através de modelos multidimensionais [Inmon 1997]. Para [Kimball e Ross 2002], a análise da base de dados transacional, permite que o analista obtenha um melhor entendimento das necessidades e limitações do projeto.

Neste momento, se fez possível definir a forma com que os componentes do projeto serão estruturados. Conforme a arquitetura *Bottom-Up*, a construção do ambiente DW é realizada através da implementação de DM independentes, desta forma o projeto tem seu desenvolvimento de forma evolutiva. Para [Kimball e Ross 2002], esta abordagem possui vantagens como, implementação e retorno rápido com enfoque inicial nos principais negócios. Na Figura 1 são ilustrados os componentes da arquitetura de aplicação de BI.

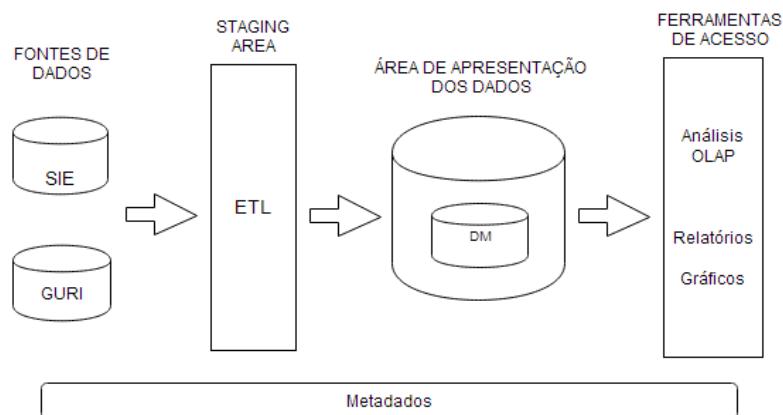


Figura 1. Arquitetura da Aplicação de BI

Com a arquitetura definida, foi iniciado o processo de criação de DM passando por quatro etapas. A primeira delas, consiste na construção do modelo dimensional a partir de um sistema transacional onde é representada a ideia central e suas dimensões, e onde são definidos como os dados serão armazenados, para permitir consultas de maneira

rápida e flexível. A etapa seguinte é o processo ETL, na qual são realizados os processos de extração, transformação e carga de dados de um sistema corporativo para um banco de dados dimensional. E, por fim, a visualização dos resultados a partir da ferramenta para interação com usuário.

3.1. Modelagem

Para a obtenção dos modelos de dados necessários para o projeto é utilizada a modelagem dimensional. A criação do modelo dimensional no processo de construção de um DM é composto pelos quatro passos, são eles: seleção do processo de negócio, definição da granularidade, escolha das dimensões e escolha dos fatos [Kimball e Ross 2002].

Desta forma, através destas etapas foi construído o modelo dimensional para esta aplicação com auxílio da ferramenta MySQLWorkbench. A Figura 2 apresenta o *Star Schema* da solução.

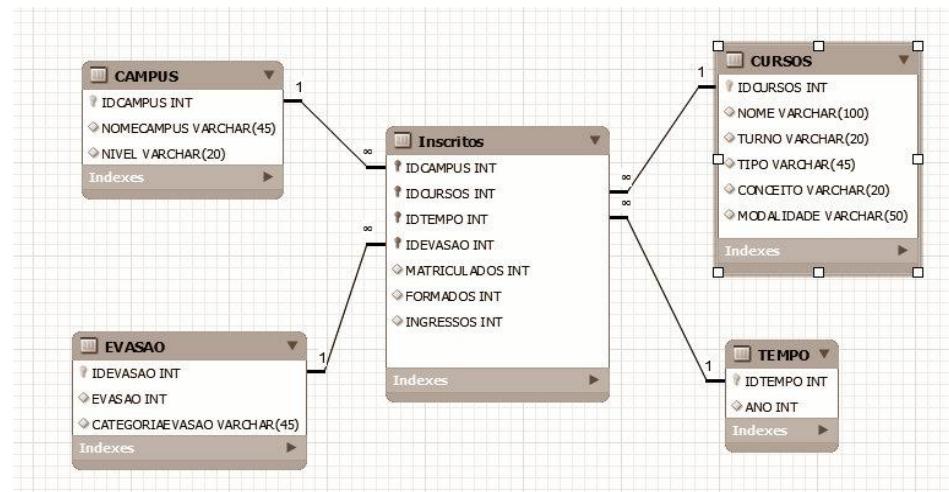


Figura 2. Star Schema

Uma vez realizada a construção do modelo de dados é possível iniciar a etapa de limpeza, transformação e carga dos dados ao repositório. Desta forma, na subseção seguinte é descrito o processo ETL.

3.2. ETL

Para migrar os dados para o DM, o primeiro passo é realizar a extração dos dados fontes, o principal objetivo desta etapa é extrair somente aqueles dados dos sistemas transacionais que serão necessários e prepará-los para as seguintes etapas do processo ETL [Cano 2007].

As bases de dados da UNIPAMPA possuem registros com todas as informações referentes as atividades acadêmicas da Instituição. Realizou-se a coleta de dados utilizados no estudo a partir do Sistema de Informação para o Ensino (SIE) da UNIPAMPA, que possui dados acadêmicos e administrativos da Instituição. Cabe ressaltar que para este trabalho a carga de dados foi limitada a um subconjunto de dados destes sistemas.

Na etapa seguinte do processo ETL os dados são pré-processados e preparados

com o objetivo de melhorar a qualidade, gerando uma base separada para a análise. Assim, os dados foram separados e organizados de acordo com as necessidades do projeto. A limpeza contribuiu para eliminar inconsistências da base como, completar dados, tratar valores nulos, e eliminar registros irrelevantes para análise.

Para este trabalho, os dados obtidos foram extraídos e preparados com auxílio da ferramenta Talend, para então serem carregados para o DM apresentando a estrutura necessária para sua utilização.

Na etapa de carga é importante garantir a correspondência entre o relacionamento das tabelas no modelo projetado. Portanto, foi necessário verificar a integridade entre chaves primárias e secundárias, para construir um ambiente analítico íntegro e confiável. Desta maneira, efetuou-se a carga segundo os modelos de tabelas fatos e dimensões, a partir do script gerado pelo diagrama do modelo construído anteriormente. Desta forma, concluída a etapa de carga dos dados para o DM é possível a visualização dos dados através das ferramentas.

3.3. Exploração dos Dados

Para iniciar o processo de análise e exploração dos dados, foram realizadas a geração de cubo OLAP para possibilitar as consultas, permitindo visões multidimensionais, a criação de gráficos e de relatórios sobre as consultas.

Para a análise OLAP os cubos foram gerados através do SpagoBI e construídos a partir das definições das tabelas fato e de dimensões. Com a estrutura básica do cubo projetada, e a modelagem dimensional previamente descrita, foram definidas para cada dimensão os atributos que seriam visualizados pelos usuários, assim como, suas hierarquias.

Na fase seguinte do processo, foi realizada a visualização dos dados tornando as aplicações de suporte à decisão entendíveis e permitindo uma melhor interpretação dos dados, para possibilitar a identificação de padrões e tendências.

Desta forma, a técnica OLAP permitiu a construção dos gráficos do sistema, possibilitando a visualização das informações estruturadas a partir dos dados existentes no sistema. As operações OLAP realizadas incluem *Slice and Dice*, visualizando dados sob diferentes pontos de vista, aplicando diferentes critérios de filtragem, *Roll Up* agregando informação detalhada a níveis superiores, e *Drill Down* possibilitando o detalhamento dos dados em níveis mais baixos.

3.4. Projeto de Dashboards

Dashboards proporcionam um mecanismo unificador que permite uma base para uma gestão eficaz e eficiente para os projetos. Proporcionam representações gráficas com as informações de negócio de maior relevância para alcançar os objetivos do analista de negócios.

A plataforma SpagoBI possui a tecnologia necessária para a construção dos painéis de controle. Desta maneira, a construção do *dashboard* realizou-se a partir do SpagoBI Studio mostrando a informação de forma útil, resumida e clara.

4. Resultados

A fim de limitar o escopo do problema e viabilizar a concentração de esforços para

conduzir a resultados significativos dentro da realidade da UNIPAMPA, e de sua atividade finalística, foi decidida a implantação de um DM acadêmico. Pode ser observado, que apesar de todos os esforços despendidos dentro da instituição, a UNIPAMPA ainda não dispõe de recursos efetivos para suporte ao processo de tomada de decisão.

No âmbito de interesse dos gestores da instituição estão as questões relacionadas ao desempenho acadêmico e evasão dos alunos em relação aos cursos. Com isto, a fim de obter uma melhor visão sobre questões relacionadas aos cursos de graduação e pós-graduação, foi realizada a construção do primeiro DM, o dm_unipampa. Este DM permitirá realizar consultas como: Avaliações sobre os cursos de graduação e pós-graduação; Análise dos cursos com maiores índices de evasão; Análise sobre número de formandos em todos os campus; Análises com o número de matrículas por campus, podendo verificar aumentos ou diminuições em relação aos últimos anos.

O processo de descoberta de conhecimento a partir de análise de dados tem por finalidade identificar conhecimentos novos e relevantes. Por outro lado, dado o conjunto de dados restrito que foi utilizado neste estudo de caso, as análises apenas apresentam padrões estatísticos, não tendo sido possível a geração de novo conhecimento.

Assim sendo, com a implementação do DM e utilização da ferramenta de *Business Intelligence* foi possível construir uma base de suporte à decisão sob uma das áreas de interesse da UNIPAMPA. Logo, para as primeiras análises construídas, foram obtidos e avaliados os resultados. Nas subseções seguintes são apresentadas algumas das análises realizadas.

4.1. Gráfico Matrículas 2012 e 2013

A Figura 3, a seguir, apresenta o gráfico construído utilizando-se os dados dos alunos matriculados nos anos de 2012 e 2013 referente a todos os campus da UNIPAMPA.

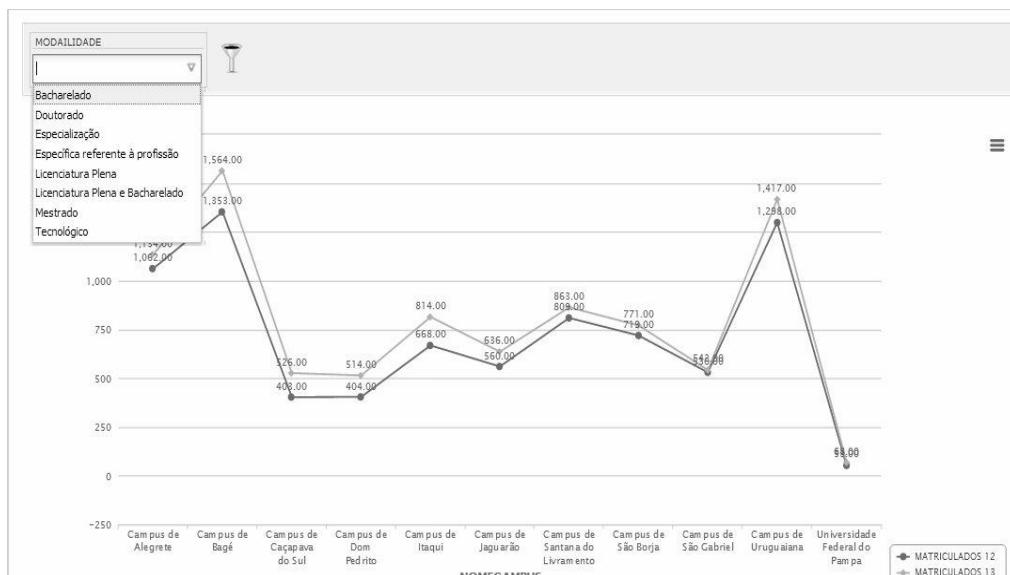


Figura 3. Gráfico Matrículas 2012 e 2013

Com o gráfico, é possível visualizar o aumento no número de matrículas em todos os campus de 2012 para 2013. Filtrando por modalidade, foi possível observar que em relação aos Bacharelados o campus Bagé foi o que obteve maior aumento no número de

matrículas. Enquanto que em relação ao número de matriculados em mestrados, foi constatada uma diminuição no campus Alegrete e um aumento considerável no campus Caçapava do Sul.

4.2. Formandos 2012

Com o gráfico apresentado na Figura 4, a seguir, é possível verificar que o Campus Uruguaiana apresentou o maior número de alunos formandos em 2012 em cursos de graduação e pós-graduação, seguido de Bagé e São Borja. Neste mesmo ano, o campus Caçapava do Sul obteve o menor número de alunos formados nestes cursos.

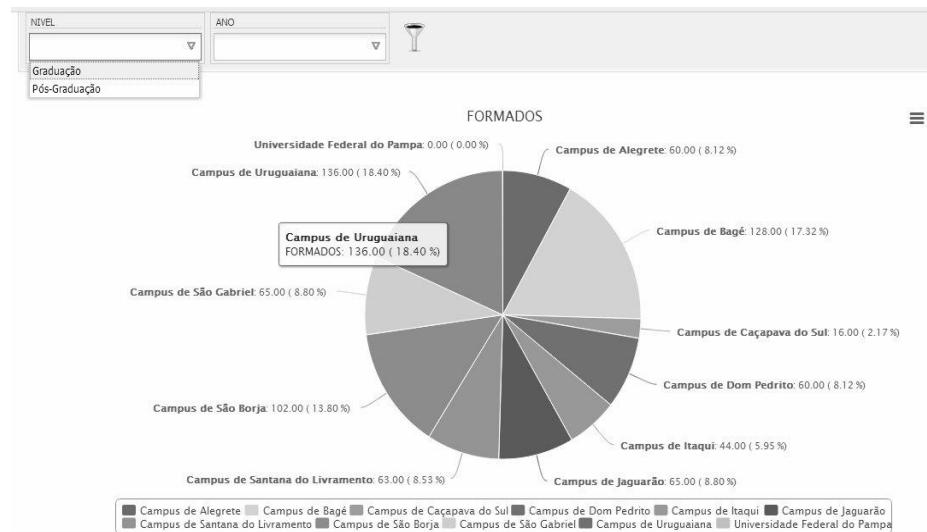


Figura 4. Gráfico Formandos 2012

Com o gráfico é possível verificar que o Campus Uruguaiana apresentou o maior número de alunos formandos em 2012 em cursos de graduação e pós-graduação, seguido de Bagé e São Borja. Neste mesmo ano, o campus Caçapava do Sul obteve o menor número de alunos formados nestes cursos.

4.3. Dashboard

4.4. Os dashboards permitem uma apresentação visual das informações mais importantes, possibilitando ao usuário avaliações e análises necessárias para alcançar os objetivos de negócio. Na Figura 5 apresenta-se um exemplo de Dashboard construído com base nos atributos números de alunos inscritos, número de alunos formados por campus e número de alunos evadidos de cada curso da UNIPAMPA.

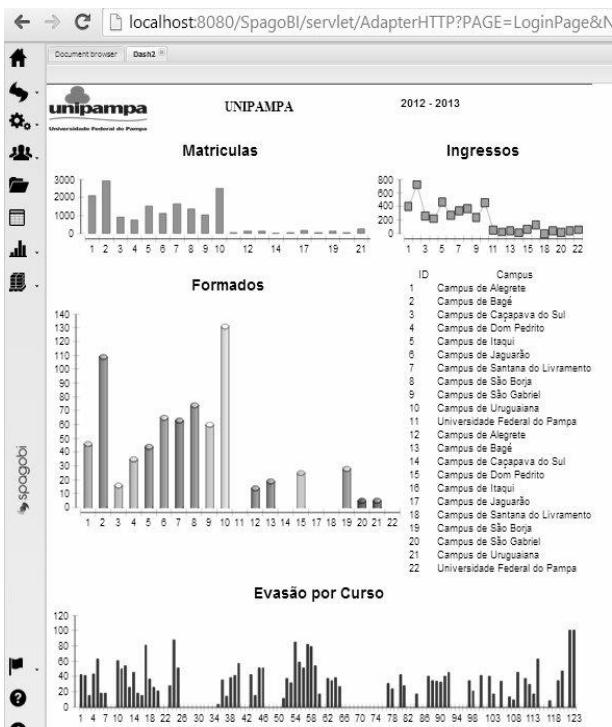


Figura 5. Dashboard UNIPAMPA

Com os resultados obtidos, é possível verificar a importância da implementação de BI na UNIPAMPA. Este tipo de tecnologia possibilita a otimização da entrega de informação, de forma completa, correta, consistente, oportuna e acessível. Facilita o processo de tomada de decisão, oferecendo um maior suporte à informação, possibilitando decisões mais rápidas e maior entendimento do impacto destas decisões. Com isto, as direções de pesquisas futuras incluem questões como: adição de informações ao modelo já construído, afim de mantê-lo atualizado para brindar mais informações e construção de outros *dashboards* para mostrar outros aspectos da DM com dados acadêmicos.

5. Referências

- Cano, J. L. (2007) "Business Intelligence: Competir con Información". Barcelona.
- Diaz, J. C. e Caralt, J. C. (2012) "Introducción al Business Intelligence". Editorial UOC. p.17-28, 51-93. Barcelona.
- Inmon, W. H. Como Construir o Data Warehouse. Rio de Janeiro, 1997.
- Júnior, M. C. (2004) "Projetando Sistemas de Apoio à Decisão Baseados em Data Warehouse". 1. ed. Rio de Janeiro: Axcel Books, v. 1.
- Kimball, R. e Ross, M. (2002), The Data Warehouse Toolkit. "The Complete Guide to Dimensional Modeling". 2.ed. John Wiley and Sons Inc.
- Turban, E. et. al. (2009) "Business Intelligence: Um enfoque gerencial para a inteligência do negócio". Bookman.
- Wu, L. et. al. (2007) "A Service-oriented Architecture for Business Intelligence". In: IEEE International Conference on Service-Oriented Computing and Applications(SOCA'07).

Uma DSL de Engenharia Reversa para Modelagem de Banco de Dados Relacionais e Geográficos Baseado em SQL/SFS

João Victor Guinelli¹, Carlos Eduardo Pantoja²

¹CEFET/RJ - UnED Nova Friburgo

Av. Gov. Roberto da Silveira, 1900 – Prado – 28.635-000 – Nova Friburgo – RJ – Brasil

²CEFET/RJ - UnED Maria da Graça

Rua Miguel Ângelo, 96 – Maria da Graça – 20.785-220 – Rio de Janeiro – RJ – Brasil

jvguinelli@gmail.com, pantoja@cefet-rj.br

Abstract. This paper presents a Domain-Specific Language for reverse engineering database systems which uses a generic metamodel for relational and geographical databases. The developed tool for Eclipse uses an integrated set of tools in order to automatically generate conceptual models using Entity-Relationship models and the OMT-G model.

Resumo. Este artigo apresenta uma Linguagem de Domínio Específica para engenharia reversa de sistemas de banco de dados que utiliza um metamodelo genérico para banco de dados relacionais e geográficos. A ferramenta desenvolvida para o Eclipse usa um conjunto de ferramentas integradas com o objetivo de gerar automaticamente modelos conceituais usando o modelo Entidade-Relacionamento e o OMT-G.

1. Introdução

A modelagem de banco de dados consiste em criar um modelo abstrato de determinado domínio apoiada por uma metodologia ou uma linguagem de modelagem. Em seguida este modelo conceitual é evoluído para um projeto lógico que resultará no projeto físico de banco de dados. O projeto físico irá gerar efetivamente o banco de dados para ser implementado em algum SGBD [Navathe; Elmasri, 2005]. Contudo, ao se deparar com bancos de dados em funcionamento em uma organização, muitas vezes este sofre alterações e por descuido o modelo conceitual acaba não sendo atualizado. Nesses casos é possível utilizar a engenharia reversa, para, a partir do *script* em SQL/SFS (Simple Feature Specification), gerar o modelo conceitual correspondente.

Existem algumas ferramentas que realizam a engenharia reversa para a modelagem conceitual a partir de uma base de dados como o MySQL Workbench [MySQL Workbench, 2015], que gera modelos em Crow's Foot [Halpin, 1999] e é embutido no SGBD; e o Visio Professional [Visio Professional, 2015], que é uma ferramenta proprietária e também gera modelos baseados em Crow's Foot. Porém tais ferramentas não flexibilizam na escolha da linguagem de modelagem, obrigando o projetista a se adequar à ferramenta e em ambos os casos não geram modelos geográficos. O objetivo deste trabalho é realizar a engenharia reversa de uma codificação em SQL/SFS para geração automatizada de uma modelagem conceitual para banco de dados relacionais ou geográficos.

Para isso, é preciso que seja possível inferir os conceitos de uma linguagem de modelagem a partir do *script* SQL/SFS. Então, será usado o *Xtext*, que é um *framework*

para desenvolvimento de *Domain-Specific Languages* (DSL). Como domínio da aplicação será usado o *Generic Database Metamodel* (GEDBM) [Guinelli et al., 2014], que integra em um único meta-modelo conceitos geográficos e relacionais, além de ser assistido por uma ferramenta *Model-Driven Architecture* (MDA) para linguagem de modelagem ER [Rosa; Pantoja, 2013]. Como o *Xtext* gera uma *Abstract Syntax Tree* (AST), que é um modelo que representa o que está sendo descrito pela linguagem SQL, é realizado um mapeamento entre este modelo e o GEDBM utilizando a *Query-View-Transformation* (QVT), que é uma especificação para transformações entre modelos [OMG, 2011]. Também é necessária a realização de um mapeamento entre o modelo GEDBM e o modelo OMT-G utilizado pela ferramenta OMT-G Design [Martinez; Frozza, 2014], tal ferramenta é utilizada para a exibição do modelo geográfico equivalente ao SQL/SFS fornecido. A ferramenta desenvolvida é um *plug-in* para o ambiente eclipse.

Este artigo está estruturado da seguinte forma: a seção 2 apresenta a metodologia para engenharia reversa de banco de dados; na seção 3 um exemplo simples utilizando a DSL é apresentado; e na seção 4 é feita uma breve discussão sobre o trabalho.

2. A Metodologia Proposta

Esta seção apresenta a metodologia utilizada pela ferramenta proposta neste trabalho. A metodologia tem como o domínio da aplicação o GEDBM, que foi escolhido por oferecer tanto conceitos geográficos como relacionais em um único meta-modelo, além de possuir uma extensão geográfica para geração de codificação SQL/SFS e compatibilidade com o modelo OMT-G [Borges et al., 2001]. A ferramenta OMT-G Design foi escolhida devido ao fato dela usar uma abordagem apoiada por modelos e ser acoplável ao GEDBM. A metodologia pode ser vista na Figura 1.

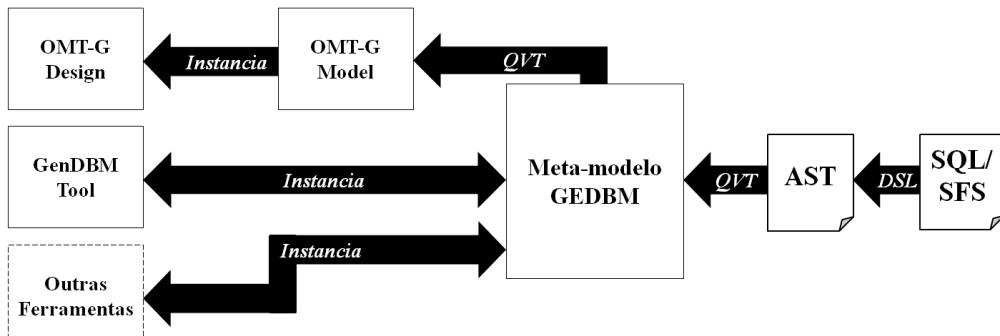


Figura 1. A engenharia reversa proposta na metodologia.

A metodologia se inicia através da codificação em SQL/SFS de uma base de dados existente. Então, a partir das construções presentes no *script* é gerada automaticamente a AST equivalente, sendo necessário em seguida executar a transformação entre esse modelo e o GEDBM. Uma vez que o GEDBM estiver instanciado é possível realizar duas opções: gerar automaticamente um modelo conceitual utilizando a GenDBM Tool para modelagens em ER; ou realizar um mapeamento entre o GEDBM e o modelo OMT-G usando o QVT. Ao final do mapeamento o modelo OMT-G estará instanciado e uma representação gráfica poderá ser gerada automaticamente. Caso exista alguma outra ferramenta de modelagem gráfica que seja aderente ao GEDBM e construída diretamente a partir dele, a construção do modelo poderá ser feita automaticamente. Caso contrário, novos cartuchos de mapeamentos deverão ser desenvolvidos.

3. Um Simples Exemplo

Esta seção apresenta um exemplo utilizando a metodologia e a ferramenta de engenharia reversa proposta. Na Figura 2 pode ser visto uma parte dos *scripts* utilizados como entrada na ferramenta e a visualização da AST que é gerada automaticamente.

The screenshot shows two windows side-by-side. On the left is a code editor window titled "ERBD.sql" containing DDL/SQL scripts for creating a database named "ERBD" with three tables: Department, Employee, and Project. The Department table has fields id (int), name (varchar(80)), and a primary key constraint on id. The Employee table has fields enrolment (int), name (varchar(80)), and foreign key constraints fk1_department and fk1_employee_id referencing the Department and Employee fields respectively. The Project table has fields id (int), description (varchar(80)), and name (varchar(80)). On the right is an "Outline" window titled "ERBD" which displays the generated Abstract Syntax Tree (AST). It shows three entities: Department, Employee, and Project, each with their respective attributes (id, name, etc.) and relationships (pk_department_ID, pk_employee_ID, fk1_department, fk1_employee_id).

```

CREATE DATABASE ERBD;

CREATE TABLE ERBD.Department (
    id int,
    name varchar(80),
    CONSTRAINT pk_department_ID PRIMARY KEY ( id )
);

CREATE TABLE ERBD.Employee (
    enrolment int,
    name varchar(80),
    id1 Department int,
    CONSTRAINT pk_employee_ID PRIMARY KEY ( enrolment ),
    CONSTRAINT fk1_department FOREIGN KEY id1_Department
        REFERENCES Department ( id )
);

CREATE TABLE ERBD.Project (
    id int,
    description varchar(80),
    name varchar(80),
);

```

Figura 2. O script DDL/SQL utilizado e a instância do AST.

Em seguida, é necessário executar a transformação que receberá como entrada o modelo AST e irá gerar como saída o modelo GEDBM instanciado. Estas regras de transformação funcionam da seguinte forma: a base de dados do *script* é transformada em um objeto *DataBase* do modelo GEDBM; a seguir, as tabelas são transformadas em objetos do tipo *Entity* do modelo de destino e o atributo *Type* destas entidades são definidos de acordo com o tipo que possuem. Em caso de banco de dados geográfico, o tipo geográfico do campo será mapeado. Já as que possuem apenas campos dos tipos comuns passam pelo mesmo processo, mas seu tipo é definido como *Conventional*; depois, os campos pertencentes às tabelas que não possuem chave primária associada são transformados em *CommonField*, já os que possuem chave primária associada são transformados em *IdentifierField*; por fim, cada chave estrangeira definida dá origem a um *Relationship* com cardinalidade de um para muitos entre a *Entity* no qual esta chave está contida e a que é referenciada por ele. Assim, quando estas transformações são executadas sobre o *script* inicial, tem-se como resultado a modelagem apresentada na Figura 3.

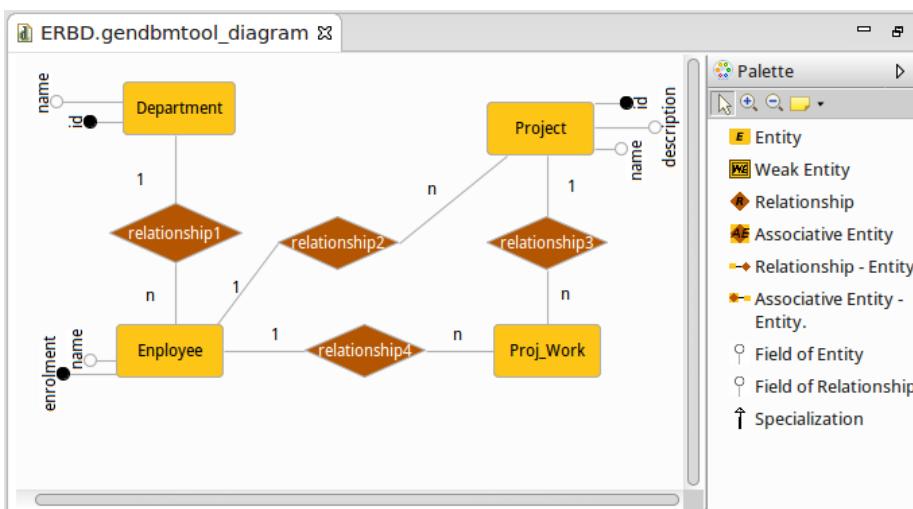


Figura 3. O modelo conceitual gerado pela ferramenta.

4. Considerações Finais

Este artigo apresentou uma DSL baseada em SQL/SFS para engenharia reversa de bancos de dados geográficos e relacionais que utiliza o GEDBM como meta-modelo central. A ferramenta também utiliza o OMT-G Design para representação de modelos geográficos e foi construída para o ambiente Eclipse através do Xtext. A DSL gerada permite que o GEDBM seja instanciado automaticamente a partir de qualquer *script* SFS /SQL. A utilização do GEDBM como domínio específico de aplicação permite que a engenharia reversa possa ser realizada para diversas linguagens de modelagens relacionais e para modelagens de banco de dados geográficos e ainda permite a adição de novas tecnologias e linguagens devido ao uso da MDA.

Ao se utilizar a ferramenta, é possível gerar automaticamente modelos conceituais que garantam a rastreabilidade dos conceitos entre o *script* e o modelo conceitual gerado, uma vez que a ferramenta utiliza especificações formais garantindo que os conceitos presentes no *script* estejam garantidamente presentes na modelagem conceitual. O objetivo da ferramenta não é ser "mais uma ferramenta de modelagem e engenharia reversa", mas sim prover em uma única solução opções ao projetista na manutenção e modelagem de um banco de dados, independente da linguagem ou abordagem usada em qualquer fase do processo de desenvolvimento de um banco de dados.

Como trabalho futuro pretende-se: expandir a abrangência da DSL para todas as construções relacionais e geográficas; desenvolver uma ferramenta gráfica para o *Crow's Foot* que instancie diretamente o GEDBM; e adicionar à metodologia o UML GeoFrame.

5. Referências Bibliográficas

- Borges, K. A. V., Davis Jr., C. A., Laender, A. H. F. (2001) "OMT-G: An Object-Oriented Data Model for Geographic Applications". Geoinformatica, v. 5, n. 3, p. 221-260, 2001.
- Elmasri, R., Navathe, S. B. (2005). "Sistemas de banco de dados". Editora Pearson.
- Guinelli, J. V., Rosa, A., Pantoja, C. E., Choren, R. (2014). "Uma Metodologia Para Apoio ao Projeto de Banco de Dados Geográficos Utilizando a MDA". Em: X Simpósio Brasileiro de Sistemas de Informação, 2014, Londrina: SBC, 2014.
- Halpin, T. (1999) "Entity Relationship Modeling from an ORM perspective: Part 1".
- Martínez, A. O. T. ; Frozza, A. A. (2014) "OMT-G Design: Uma Ferramenta para Modelagem de Dados Espaciais". In: X Escola Regional de Banco de Dados, 2014, São Francisco do Sul: 2014.
- MySQL Workbench. Disponível em: <http://www.mysql.com/products/workbench/>. Acesso em: 14/03/2015.
- OMG. (2011) "Meta object facility (MOF) Query/View/Transformation specification.". URL: <<http://www.omg.org/spec/QVT/1.1/PDF/>>.
- Rosa, A., Pantoja, C. E. (2013). "Uma Ferramenta MDA para Modelagem de Banco de Dados Relacionais". Em: IX Escola Regional de Banco de Dados, Camboriú: 2013.
- Visio Professional. Disponível em: <<https://products.office.com/pt-br/visio/visio-professional-2013-business-and-diagram-software>>. Acessado em: 13/03/2015.

ntSQL: Um Conversor de Documentos RDF para SQL

Flávio R. Bayer¹, Lucas L. Nesi¹, Rebeca Schroeder¹

¹Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – Caixa Postal 15.064 – Joinville – SC – Brasil

{flaviobayer, lucas3lnesi}@hotmail.com, rebeca.schroeder@udesc.br

Resumo. A disseminação do modelo RDF como formato de publicação de dados na Web requer novas formas de armazenamento. Sistemas NoSQL têm sido aplicados para dar suporte a fontes de dados RDF de larga escala, enquanto SGBDs relacionais vêm sendo utilizados para conjuntos de dados de menor porte. No uso de SGBDs relacionais para este propósito a ausência de um esquema RDF acarreta em uma sub-utilização das composições do modelo relacional. Este artigo apresenta ntSQL, uma ferramenta que automaticamente extrai o esquema de um documento RDF em formato NT a fim de produzir um banco de dados relacional equivalente. O artigo descreve as etapas deste processo, bem como resultados preliminares quanto ao desempenho da ferramenta.

1. Introdução

De acordo com o Projeto *Linked Open Data*¹, RDF (*Resource Description Framework*) é hoje o modelo padrão para publicação de dados na Web. O modelo RDF estende a estrutura de *links* da Web ao permitir definir semanticamente os relacionamentos e os componentes estabelecidos por um *link* através do uso de URIs (*Universal Resource Identifier*). A unidade base de definição de dados deste modelo é a tripla, composta de *sujeito* e *objeto* relacionados por um *predicado*. Um documento RDF corresponde a um conjunto de triplas que podem estar definidas através de diferentes formatos assumidos para RDF, como NT-triples (NT), RDF/XML e N-Quads. Dentre as três opções, o formato NT destaca-se por ser um modelo mais compacto que os demais[Beckett 2014].

Em geral, sistemas NoSQL e repositórios de grande escala têm sido adotados para o armazenamento de fontes RDF em virtude do elevado volume de dados que algumas fontes apresentam[Zeng et al. 2013]. Entretanto, para repositórios de dimensões inferiores o uso SGBDs relacionais são preferíveis pois evitam a complexidade de gerenciamento que acompanha sistemas NoSQL. Neste contexto, a ausência de um esquema RDF faz com que diversas soluções utilizem SGBDs relacionais na forma de *triple store*. Em um *triple store*, dados RDF podem ser armazenados como um conjunto de triplas em uma única tabela[Abadi et al. 2009]. Conforme demonstrado por Zeng et al[Zeng et al. 2013], esta forma de armazenamento além de gerar grandes arquivos para as relações, leva a operações de junção custosas. Em resumo, a ausência de um esquema, que justifica a adoção deste tipo de solução, acaba por sub-utilizar o modelo relacional ao criar relações baseadas apenas nas composições de triplas. Apesar de RDF constituir um modelo livre de esquema, observa-se que é possível a extração de estruturas de dados a partir de diversas fontes RDF[Minh-Duc and Boncz 2013]. Esta possibilidade viabiliza uma representação RDF mais adequada em modelos de bancos de dados relacionais.

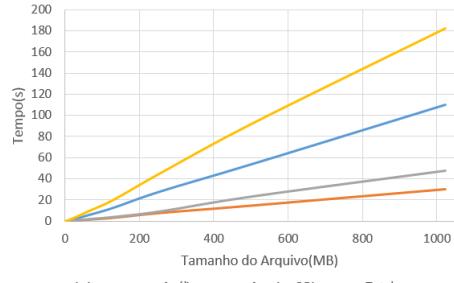
¹<http://linkeddata.org/>

```

<http://Product1> <http://type> <http://Product>.
<http://Product1> <http://rdf-schema#label> "Board".
<http://Product1> <http://rdf-schema#comment> "New".
<http://Product1> <http://rdf-schema#day> "20".
<http://Product1> <http://rdf-schema#month> "2".
<http://Product1> <http://rdf-schema#year> "2010".
<http://Product2> <http://type> <http://Product>.
<http://Product2> <http://rdf-schema#label> "Cap".
<http://Product2> <http://rdf-schema#comment> "Red".
<http://Product2> <http://rdf-schema#day> "2".
<http://Product2> <http://rdf-schema#month> "7".
<http://Product2> <http://rdf-schema#year> "2012".

```

(a) Tripas RDF em Formato NT



(b) Desempenho do ntSQL

Figura 1. Tripas RDF e Resultados Preliminares do ntSQL

Este artigo considera a possibilidade de extração de esquemas RDF, e apresenta uma ferramenta chamada ntSQL para geração de bancos de dados relacionais. A ferramenta ntSQL é capaz de extrair um esquema relacional a partir da análise de um documento RDF no formato NT e, em seguida, produzir um *script* de importação para um banco de dados relacional. A seção seguinte descreve o processo de extração de esquema e geração do *script*, em conjunto com um experimento preliminar que revela o desempenho do ntSQL. As considerações finais deste trabalho são apresentadas na Seção 3.

2. Conversão NT para SQL

Usualmente, documentos RDF utilizam a propriedade *type* para conectar entidades às suas respectivas classes. Considere as tripas RDF fornecidas pela Figura 1(a) e observe a relação estabelecida entre elas. É possível inferir que o sujeito <http://Product1> está relacionado à classe <http://Product> devido ao relacionamento estabelecido pelo predicado *type* na primeira linha. O mesmo ocorre para o sujeito com valor <http://Product2>. Desta forma é possível identificar que haverá uma relação do tipo *Product*, sendo *Product1* e *Product2* tuplas desta relação. Os predicados das demais tripas referem-se a campos desta relação (*label*, *comment*, *day*, *month* e *year*).

Desta forma, a geração de um esquema relacional a partir de um arquivo NT é um processo que basicamente consiste em verificar todos os sujeitos de tripas que podem estar na mesma relação. Esta verificação consiste em que todos os sujeitos identificados tenham o predicado *type* e um mesmo valor para o objeto, que corresponderá ao nome da relação a qual pertencem. Para as demais tripas que não estejam associadas a um predicado do tipo *type*, assume-se que corresponderão aos campos das relações identificadas.

A ferramenta ntSQL executa as etapas de leitura do arquivo NT, análise e geração do *script* SQL. Na primeira etapa, o arquivo NT é percorrido com a finalidade de extrair o esquema relacional preliminar bem como seus dados. Na fase seguinte, o esquema relacional final é obtido a partir da análise de associações entre relações para definição de relacionamentos N:N e chaves estrangeiras. Na última etapa, um *script* SQL para geração da base relacional é gerado como resultado do processo. As respectivas etapas são descritas a seguir, em conjunto com resultados preliminares do desempenho da ferramenta apresentados pela Figura 1(b). Os arquivos NT utilizados nos exemplos bem como no experimento foram obtidos a partir do gerador de documentos RDF do Berlin Benchmark².

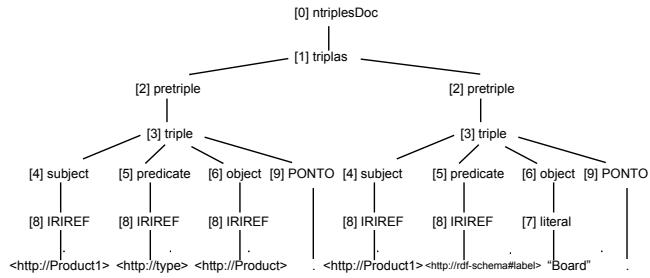
²<http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/>

```

(0) ntriplesDoc → triplas
(1) triplas → triplas pretriple | pretriple
(2) pretriple → triple | TCOMENTARIO | TEOL
(3) triple → subject predicate object PONTO
(4) subject → iriref | BLANK_NODE_LABEL
(5) predicate → iriref
(6) object → iriref | BLANK_NODE_LABEL | literal
(7) literal → STRING_LITERAL_QUOTE |
STRING_LITERAL_QUOTE SOBRECARGA iriref|
STRING_LITERAL_QUOTE LANGTAG
(8) iriref → IRIREF
(9) PONTO →
(10) SOBRECARGA → ^^
(11) LANGTAG → @[a-zA-Z]+(-[a-zA-Z0-9]+)*
(12) EOL → [\n\r]+
(13) IRIREF → <[^#x00-\#x20<>{}|^`\\]>
(14) BLANK_NODE_LABEL → _:[:a-zA-Z0-9]*_
(15) ECHAR → \[tbnrf"\`]
(16) COMENTARIO → #(\^`n\z)*
(17) STRING_LITERAL_QUOTE →
"([^\#x22#\x5C#\xA#\xD] | ECHAR)*"

```

(a) Gramática aplicada por ntSQL



(b) Árvore de Derivação

Figura 2. Leitura de Arquivos NT

2.1. Etapas de Processamento do ntSQL

As regras de leitura do arquivo NT foram criadas com base na gramática disponibilizada pela documentação da W3C[Beckett 2014]. As alterações efetuadas na gramática se resumem à aceitação de comentários e à remoção de caracteres *UNICODE*. A Figura 2(a) apresenta a gramática aplicada pelo ntSQL. O documento NT deve respeitar esta gramática e estar no formato *ASCII*. Algumas outras restrições foram criadas por motivos de implementação, como o limite máximo não exceder o tamanho de página do S.O. usado, visto que o programa desenvolvido utiliza apenas a memória principal. A fim de demonstrar o funcionamento da gramática, considere as duas primeiras triplas da Figura 1(a). Ao iniciar da produção 0, são aplicadas sucessivas derivações de produções até ocorrer o reconhecimento total da entrada. A Figura 2(b) apresenta a árvore gerada para o reconhecimento desta entrada.

Um *parser* foi criado usando as ferramentas GNU Bison³ e Flex⁴. O Flex é uma ferramenta para geração de *scanners* que são usados para fazer o reconhecimento de padrões léxicos em um texto. Os padrões são processados como expressões regulares definidas. Neste caso, as produções 9 a 17 foram utilizadas. No final do processo o Flex gera um código fonte em C que será futuramente usado pelo Bison para reconhecimento da entrada. No caso da Figura 2(b), o Flex é o responsável pelo reconhecimento dos *tokens* terminais, representando as folhas da árvore. O GNU Bison é um gerador de reconhecedores de linguagem, que converte uma gramática livre de contexto em um reconhecedor de linguagens LR (*Left to Right*) determinísticas, usando tabelas de reconhecimento LALR (*Look-Ahead LR parser*). O Bison recebeu as produções 0 a 8 e o código fonte gerado pelo Flex e como resultado final gera um código fonte em C que é um reconhecedor da gramática. No caso da Figura 2(b), o Bison tem função de realizar as reduções referentes a nós internos da árvore.

Uma leitura sequencial do arquivo NT é realizada a fim de extrair seu esquema relacional na forma de um dicionário de dados, e um mapa de sujeitos que relaciona todos os dados (instâncias) associados a ele. Sempre que ocorrer uma redução para (3), o programa verifica se o predicado corresponde a *type* e, neste caso, o nome do objeto corresponderá a uma relação do dicionário de dados. Para cada sujeito processado, o programa armazena os predicados e objetos reconhecidos no mapa de sujeitos. Após o processamento

³<http://www.gnu.org/software/bison/>

⁴<http://flex.sourceforge.net/>

de todas as triplas de um sujeito, as informações de cardinalidade, tamanho máximo do campo e tipo do campo são atualizadas no dicionário de dados.

Na etapa de Análise, quando um mesmo sujeito tem um predicado repetido e os objetos em questão corresponderem a outras relações, identifica-se uma relação N:N que deve ser armazenada em uma nova tabela com os campos ID do sujeito e do objeto, em conjunto com os demais campos identificados para esta relação. Para todas as relações identificadas, uma outra verificação é realizada para cada campo a fim de determinar se ele constitui uma referência a outro sujeito (no caso [8] *iriref* ser reduzido para [6] *object*). Em caso afirmativo, ele marca o campo da relação como uma chave estrangeira no dicionário de dados.

Um *script SQL* é gerado como produto final da leitura de um arquivo NT. Os comandos de definição de tabelas são extraídos a partir do dicionário de dados gerado, enquanto os comandos de inserção de dados são obtidos a partir do mapa de sujeitos.

2.2. Desempenho da Ferramenta ntSQL

Para realizar os testes de desempenho do programa foi utilizado uma máquina com processador Intel Core2 Duo T5200 (1.60 GHz), 2GB de memória RAM, e sistema operacional Ubuntu 14.04. Foram realizados testes com diferentes tamanhos de arquivos de entrada, variando de 1MB até 1024MB. A variação de triplas no arquivo varia proporcionalmente ao tamanho do arquivo, independentemente do esquema relacional final, sendo que para cada MB de entrada equivale a aproximadamente 3.920 triplas. Ou seja, no teste realizado foram avaliadas o número de triplas variando de 3.920 a 4.014.080. Os resultados observados são apresentados na Figura 1(b). O gráfico mostra o tempo gasto nas 3 etapas do processamento e também o tempo total acumulado. Pode-se perceber que em todas as etapas do programa o tempo necessário cresce de forma proporcional ao tamanho do arquivo e ao número de triplas. A utilização de memória pelo programa também cresce proporcionalmente ao aumento do tamanho do arquivo, para cada 1MB do arquivo, é ocupado 1,3MB na memória para seu armazenamento em sua estrutura. Por este motivo o tamanho máximo do arquivo foi de 1GB, sendo que para arquivos maiores que 1GB o tamanho da memória principal não era suficiente.

3. Considerações Finais

O artigo contribui com uma ferramenta para a conversão de arquivos RDF em formato NT para SQL. Embora em estado preliminar, a extração de esquemas RDF desempenhada pela ferramenta pode ser adaptada para a geração de esquemas conceituais capazes de servir como referencial para a geração de bancos de dados de diversos modelos. Dentre os trabalhos futuros, destaca-se o aprimoramento das funcionalidades e desempenho da ferramenta, bem como a conversão de consultas SPARQL para SQL e a comparação com *triple stores* no desempenho em consultas.

Referências

- Abadi, D. J., Marcus, A., Madden, S. R., and Hollenbach, K. (2009). SW-Store: A Vertically Partitioned DBMS for Semantic Web Data Management. *The VLDB Journal*, 18(2):385–406.
- Beckett, D. (2014). RDF 1.1 N-Triples: A line-based syntax for an RDF graph. <http://www.w3.org/TR/n-triples/>.
- Minh-Duc, P. and Boncz, P. (2013). Self-organizing Structured RDF in MonetDB. In *Proceedings of the ICDE PhD Symposium*.
- Zeng, K., Yang, J., Wang, H., Shao, B., and Wang, Z. (2013). A Distributed Graph Engine for Web Scale RDF Data. *VLDB Endowment*, 6(4):265–276.

Módulo para *Business Intelligence* Compatível com Dispositivos Móveis para o Software de Gestão Empresarial Elementare

Ismael Martiny, Helena Graziottin Ribeiro

Centro de Ciências Exatas e Tecnologia – Universidade de Caxias do Sul (UCS)
Rua Francisco Getúlio Vargas, 1130 – 95.070-560 – Caxias do Sul – RS – Brasil

imartiny@ucs.br , hgrib@ucs.br

Abstract. This paper describes the development of a Business Intelligence web application compatible with mobile devices. The advancement of these types of devices allowed enterprises to have quick and convenient access to the information available from anywhere. The information available in analytical tools through graphics makes better the decision making.

Resumo. Este artigo descreve o desenvolvimento de uma aplicação web Business Intelligence compatível com dispositivos móveis. O avanço destes tipos de dispositivos permitiu às empresas ter acesso rápido e conveniente das informações disponíveis a partir de qualquer lugar. A informação disponível em ferramentas analíticas através de gráficos torna melhor a tomada de decisões.

1. Introdução e Trabalhos Relacionados

Com o avanço das tecnologias adotadas pelas empresas atualmente, seja em hardware ou software, chega-se num estágio onde muitas delas possuem muitos dados armazenados, tendo “problemas” para explorá-los e armazená-los de forma organizada. Para tanto, cada vez mais os sistemas de gestão adotados pelas empresas tem como objetivo tornar estes dados úteis e ajuda-las nas tomadas de decisão. O objetivo do trabalho é demonstrar o desenvolvimento de uma ferramenta analítica em um novo contexto dos dispositivos móveis, de modo a ser obtido o melhor proveito respeitando as limitações deste tipo de dispositivo.

Segundo Primak (2008) a informação é fundamental para a construção do conhecimento. Portanto, a informação não é conhecimento, mas sim componente deste. Historicamente, de acordo com Primak (2008), nos anos 80 iniciou-se a aplicação do termo *Business Intelligence* (BI) que tem por objetivos segundo Turban et al. (2009) os principais objetivos do BI é o acesso interativo aos dados muitas vezes em tempo real, proporcionando a manipulação desses dados e fornecer aos gerentes e analistas de negócio a capacidade de realizar a análise adequada. Além disso, Turban et al. (2009) diz que o processo de BI baseia-se na transformação de dados em informações, posteriormente em decisões e por fim em ações.

O software Elementare Gestão Empresarial¹ possui diversos módulos para facilitar a consolidação e padronização de processos e dados das empresas, fornecendo visibilidade das informações de forma integrada. Ele possui várias rotinas de inserção de dados, sendo as mais básicas cadastros de clientes, fornecedores e produtos. Também conta com algumas rotinas mais complexas que contemplam a questão financeira e gerencial através de gráficos e tabelas dinâmicas, bastante úteis no processo de BI. Apesar do sistema já contar com vários tipos de gráficos e consultas, notou-se através de solicitações dos clientes a necessidade de ampliar estes tipos de consultas, a fim de obter um maior cruzamento das informações inseridas e tê-las disponíveis a qualquer momento, em especial nos dispositivos móveis.

No mercado já existem algumas soluções de empresas tais como a SAP² e Tableau Software³ que oferecem aos seus clientes soluções móveis para *Business Intelligence* para facilitar a tomada de decisão dos gestores das empresas de maneira rápida e fácil com opção *on-line* e *off-line*. Estas ferramentas contemplam uma base de dados on-line onde são apresentadas em formas de gráficos e *dashboards*, tais como: gráfico geográfico com o acumulado de vendas, resumo e posição da empresa em relação às metas estabelecidas. Podem ser acessadas de qualquer lugar sem a necessidade de compra de hardware específico.

2. Ferramentas de BI e Dispositivos Móveis

Para Davenport e Harris (2007) a inteligência analítica é um subsistema do que passou a ser chamado de *business Intelligence*. Assim, ela compreende a utilização de dados, e a análise qualitativa e estatística com gestão baseada em fatos para a tomada de decisões e ações. Podemos dizer que o BI é amplo no aspecto organizacional e que é utilizado também para fornecer indicadores financeiros por meio de simulações e de cenários com previsões futuras.

Para esta análise descrita anteriormente, principalmente os gráficos e *dashboards* auxiliam os gestores em suas decisões. Eles estão utilizando cada vez mais seus dispositivos móveis para acesso aos seus e-mails, tarefas e compromissos, notícias e acompanhamento de sua empresa. Contudo os dispositivos móveis possuem algumas características e limitações segundo Weyl (2014). Os aplicativos usados nos dispositivos, tais como navegadores de internet possuem recursos limitados quanto ao tamanho da tela. Devido à tela e à resolução ser menor, os aplicativos e sites devem ser adaptados para uma melhor visualização.

Atualmente o que se tem visto é o aumento do desenvolvimento de sites com *layout responsivo*. Isto significa que independente do tamanho da tela do dispositivo, o site ou conteúdo visualizado adapta-se a esse tamanho. Isso ocasiona uma melhor disposição dos objetos sem que se crie barra de rolagem horizontal. Esta ideia é baseada na experiência do usuário com as telas sensíveis ao toque, já que é muito fácil rolar o conteúdo da tela com o dedo de baixo para cima ou vice-versa.

1 <http://www.elementare.inf.br/>

2 www.sap.com

3 www.tableau.com

A disponibilização de uma ferramenta analítica para os dispositivos móveis permite mostrar um panorama geral da empresa com resumos de vários de seus setores, praticamente numa tela só. Isso é resolvido através do que chamamos de *dashboard*, onde há uma grande concentração de gráficos e indicadores que facilitam a visualização das informações com uma percepção rápida da situação pelos gráficos apresentados.

3. Aplicativo Elementare BI Móvel

A solução para o projeto foi definida num conjunto de recursos de visualização, adequados aos dispositivos móveis que consegue independente do tamanho de tela ou sistema operacional. Está disponível nas principais plataformas móveis do mercado tais como, iOS, Android e Windows Phone. É para web, e o usuário poderá utilizar o navegador que está habituado para acessar a internet. A vantagem deste tipo de solução é a abrangência de dispositivos, e pelo desenvolvimento do módulo uma única vez centralizando e facilitando a liberação de futuras atualizações. A arquitetura é baseada nos seis elementos de Davenport e Harris (2007) onde só não houve tanta ênfase no aspecto dos processos operacionais.

A interface que o usuário possui no dispositivo móvel está em formato responsivo que pode ser acessado em qualquer plataforma móvel e em diversos tamanhos de tela. As telas responsivas se ajustam automaticamente de tamanho reformulando os menus para uma melhor disposição dos elementos em telas de dispositivos menores. O exemplo de uma tela do Elementare BI observa-se na Figura 1 que está redimensionada conforme o tamanho da tela do dispositivo.

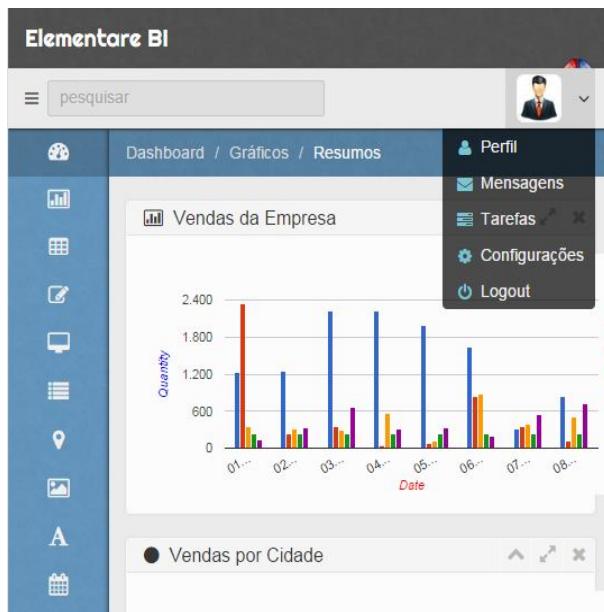


Figura 1. Layout do Elementare *Business Intelligence*

Para que o *Business Intelligence* compatível para dispositivos móveis fosse colocado em prática, foi necessário a construção de um processo completo baseado em *Data Warehouse*. A seleção de dados foi efetuada através do banco de dados do sistema Elementare Gestão Empresarial. Além disso, foi necessário o desenvolvimento de um algoritmo de extração, transformação e carga (ETL) que populou o banco de dados da

aplicação *web*. O algoritmo de ETL foi desenvolvido em cima da estrutura do sistema Elementare Gestão Empresarial, tendo uma rotina exclusiva que é executada e pré-configurada pelos usuários administradores para a carga dos dados no *Data Warehouse* empresarial. Esta rotina é configurada e executada de maneira automática ou manual conforme a necessidade de cada empresa que utiliza a ferramenta.

O banco de dados que foi populado pelo processo de ETL é um banco de dados MySQL que está hospedado na nuvem para estar acessível de qualquer lugar quando os usuários farão suas consultas e visualizações de informações através dos dispositivos móveis.

A camada *frontend*, através da qual os usuários realizam suas consultas e visualizações, são baseadas num framework para desenvolvimento responsivo chamado *Bootstrap*⁴ com bibliotecas de gráficos *Morris Charts* e *Google Charts*. Estas bibliotecas permitem a geração de vários tipos de gráficos configuráveis de acordo com os parâmetros enviados, e que retornam os gráficos montados. Na figura 1 observa-se o *dashboard* com os gráficos resultantes utilizando a biblioteca *Google Charts*.

As funcionalidades das soluções de BI já existentes mencionadas na introdução também podem ser encontradas no Elementare BI. A diferença destas soluções para o Elementare BI é que não há a necessidade de ter um banco de dados prévio com a versão desktop destes sistemas para posterior publicação num banco de dados em nuvem.

4. Considerações Finais

Com a adoção cada vez maior dos dispositivos móveis nas empresas, tanto por parte dos gestores como dos funcionários há a necessidade de respostas e análises rápidas partindo das ferramentas analíticas que podem ser disponibilizados nestes dispositivos. O software ainda não está finalizado e está em fase de testes com usuários de algumas empresas piloto. Algumas funcionalidades tais como, gráficos drill-down, detalhes do envio de relatórios por-email ainda não estão concluídos. Como projeto futuro e de ampliação, pensa-se uma versão *off-line* para os principais sistemas operacionais móveis disponíveis.

Referências

- Davenport, H. T. e Harris, G. J. (2007) “Competição analítica: vencendo através da nova ciência”. Rio de Janeiro: Elsevier.
- Primak, F. (2008) “Decisões com B.I. (Business Intelligence)” 1^a. ed. Rio de Janeiro: Editora Ciência Moderna.
- Turban, E. et al. (2009) “Business Intelligence: Um enfoque gerencial para a inteligência do negócio” Porto Alegre: Bookman.
- Weyl, E. (2014) “Mobile html5” São Paulo: Novatec.