

SOFTWARE DE CONVERSÃO DE BANCOS DE DADOS RELACIONAL PARA BANCO DE DADOS ORIENTADOS A GRAFO

Augusto Batista de Jesus¹, Pedro Henrique Pinheiro Leite¹, Walisson Pereira de Sousa²

¹ Discente do Curso Técnico em Informática ao Ensino Médio – IFTO Campus Araguaína, e-mail: augustosousa790@gmail.com; pedro.lpinheiro3112@gmail.com

² Docente do Curso Técnico em Informática ao Ensino Médio – IFTO Campus Araguaína, e-mail: walison.sousa@gmail.com

Resumo: Converter uma base de dados nem sempre foi algo fácil, devido à falta de um software capaz de realizar essa conversão, com falta de um software capaz de converter, realizar essa tarefa se tornou algo bastante difícil. A intenção de realizar uma conversão do tipo de banco de dados surge na necessidade de migrar para uma base de dados com uma capacidade de guardar um grande número de dados, uma dessas bases que suporta um número excessivo de dados está presente no modelo NoSQL, conhecida como base Orientada a Grafos. Esse trabalho tem a finalidade de desenvolver um *software* capaz de realizar uma conversão de bancos de dados, capaz de converter o banco de dados presente na forma relacional para a orientada a grafos, usando a linguagem Java em conjunto com a API JGraph, visando resolver a dificuldade de se converter uma base de dados.

Palavras-chave: Converter, Grafos, Java, JGraph, *Software*

1 INTRODUÇÃO

Os bancos de dados em que orientados a grafos podem ser caracterizados como aqueles nos quais as estruturas de dados em que são modeladas na forma de grafos, e a manipulação de dados é expressa através de operações orientadas a grafos. Nesse tipo de estrutura, os dados são representados por nós, arestas e propriedades. Os nós representam as entidades, as arestas expressam as relações entre os nós e as propriedades apresentam características das entidades e relacionamentos (ALVAREZ et al., 2016).

Os primeiros trabalhos em teoria dos grafos surgiram no século XVII. Vários autores publicaram artigos neste período, com destaque para o problema descrito por Leonhard Euler, conhecido como as Pontes de Königsberg. Esse problema foi exposto em um artigo publicado em 1736, na cidade de Königsberg, Alemanha. Sete pontes estabeleceram ligações entre as margens do Rio Pregel e duas de suas ilhas. Para resolver o problema, deveria-se, partindo de um determinado ponto, passar por todas as pontes somente uma vez e retornar ao ponto inicial, como exemplificado pela Figura 1. Outro grande pesquisador foi o alemão Kirchhoff que utilizou grafos para modelagem de circuitos elétricos, já no século XIX (CARVALHO, 2005).

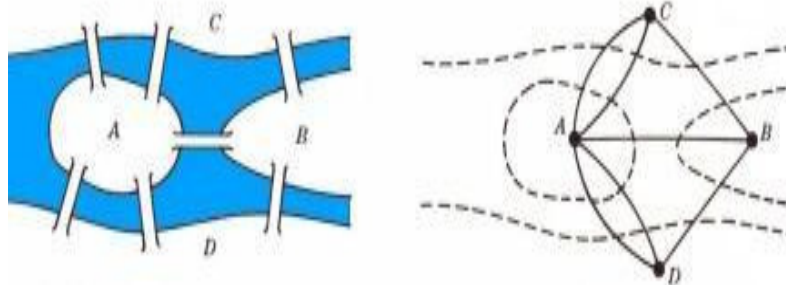


Figura 1 - As Pontes de Königsberg Fonte: Helder Rodrigues - Acervo Pessoal

Nos últimos anos, com o advento da *Web Semântica* e a popularização do movimento NoSQL (*Not Only Structured Query Language*), a adoção de modelos de bancos de dados de grafos está se tornando cada vez mais comum. Este modelo se particularmente apropriada se quando o modelo de dados não possui uma estrutura rígida ou apresenta grandes volumes de relações, pois têm uma infraestrutura otimizada para responder a operações com ênfase nas relações, por exemplo, percorrer um caminho, realizar uma busca em largura ou em profundidade. Mesmo com esse crescimento na utilização de bancos de dados de grafos, temos um grande legado em todas as áreas de sistemas que utilizam o modelo relacional para armazenamento dos dados, e este modelo é a melhor opção para alguns tipos de problema (CAVOTO, 2014).

Os grafos estão presentes nos mais diversos sistemas e situações, como por exemplo nos sistemas de navegação, nas redes sociais como o Facebook e Twitter, nos sistemas de reconhecimento biométrico, no controle de tráfego de ruas e avenidas, entre outros. Até mesmo pesquisas sobre o *DNA (Deoxyribonucleic Acid)* têm sido beneficiadas com a utilização da teoria dos grafos.

Converter uma base de dados para outro servidor de banco de dados sempre foi visto como uma tarefa complicada, embora converter uma base de dados não seja ideal, mas uma necessidade (DIANA, 2010).

Os bancos de dados relacionais não permitem que os dados sejam representados através de grafos de uma maneira natural, com isso, algumas pesquisas se tornam extremamente complexas, ou até mesmo impraticáveis nos bancos de dados relacionais. Em contrapartida, com a popularização dos bancos de dados não relacionais e as diferentes formas de persistir dados, começam a surgir alternativas para a situação de dados que devem ser persistidos e consultados através de uma representação em grafo (ALMEIDA, 2011).

Este projeto tem como objetivo o desenvolvimento de um *software* capaz de converter o modelo de um banco de dados relacional para um modelo orientado a grafos, ajudando o usuário na conversão de modelos e migração dados. Após a conversão, o usuário poderá observar o banco de dados orientado a grafos gerado, de modo que o mesmo compreenda o conteúdo presente, facilitando a pesquisa e a leitura do modelo relacional na forma de grafos.

2 METODOLOGIA

O projeto desenvolvido tem o intuito de realizar a conversão de bancos de dados relacionais para o modelo orientado a grafos, utilizando na linguagem Java SE (*Standard Edition*) para a criação da aplicação, juntamente com a API JGraph. Esta última, também desenvolvida na linguagem Java, é utilizada para a criação/desenho dos elementos visuais do grafo, o programa usado no desenvolvimento do *software* será a IDE NetBeans.

3 MATERIAS UTILIZADOS

3.1 Programas Utilizados

Para a elaboração deste projeto utilizou-se a NetBeans IDE que utiliza a linguagem Java.

a) NetBeans IDE é o IDE oficial para Java 8. Com seus editores, analisadores de código e conversores, você pode atualizar rapidamente e sem problemas seus aplicativos para usar novas construções de linguagem Java 8, como lambdas, operações funcionais e referências de métodos (NETBEANS, 2017).

3.2 Linguagens Utilizadas

Quanto à linguagem de programação utilizada nesse projeto, foi utilizada as seguintes:

a) A linguagem Java é utilizada em todos os tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis e incorporadas, jogos, conteúdo baseado na *Web* e *softwares* corporativos. Com mais de 9 milhões de desenvolvedores em todo o mundo, de forma eficiente, o Java permite que você desenvolva, implante e use aplicações e serviços estimulantes (JAVA,2017). Para o desenvolvimento desta aplicação, dentro do conjunto de versões da linguagem disponíveis, foi escolhida a versão *Standard Edition*, o padrão utilizado para aplicações *Desktop*.

Juntamente com a linguagem Java, a API JGraph também foi utilizada. Esta possui um controle que exibe uma rede de objetos relacionados usando o paradigma bem conhecido de um grafo. O JGraph exibe seus dados, desenhando elementos individuais, como exemplo a figura.

Cada elemento exibido pelo gráfico contém exatamente um item de dados, que é chamado de célula. Uma célula pode ser um vértice ou uma aresta. Os vértices podem ter vizinhos ou não, e as arestas podem ter vértices fonte e alvo ou não, dependendo se eles estão conectados (JGRAPH,2017).

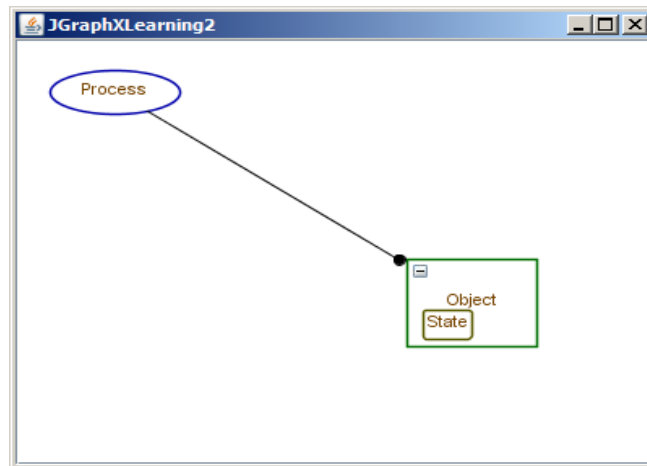


Figura 2 - Interface JGraph Fonte: Vainolo's Blog - Acervo Pessoal

b) O SQL (*Structured Query Language*) é uma linguagem de consulta a banco de dados criada na década de 70 na IBM, com base em conceitos da Álgebra relacional. Foi desenvolvida para padronização dos sistemas gerenciadores de banco de dados relacionais existentes. (CÉSAR, 2012).

c) A UML – *Unified Modeling Language* ou Linguagem de Modelagem Unificada – é uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de Orientação a Objetos. É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem-padrão de modelagem adotada internacionalmente pela indústria de Engenharia de *Software* (GUEDES, 2011).

4 RESULTADOS E DISCUSSÃO

O sistema discutido neste trabalho encontra-se em desenvolvimento. No entanto, nesta seção, serão apresentados os resultados parciais do processo de criação do sistema. Esta seção apresenta resultados parciais como imagens do sistema em questão, diagramas de classes de que explica as principais funções do sistema e o modo em que ele opera.

Como agregado do processo de desenvolvimento:

Os softwares de conversão de banco de dados estão divididos em telas, como as que são citadas a seguir:

A Tela de conexão com o banco de dados (figura 3), em que o usuário conectará em seu servidor, a porta em que se localiza o servidor. O usuário terá que fazer essa conexão com o nome de usuário a senha do servidor de banco de dados

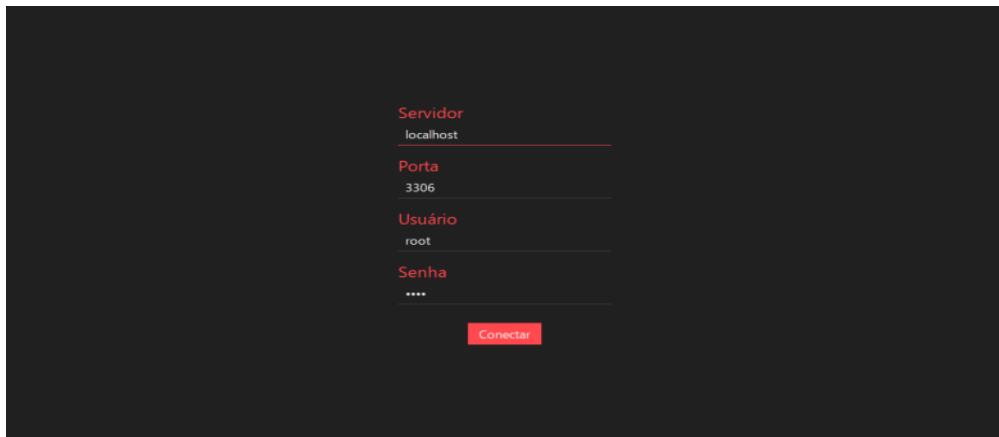


Figura 3 - Tela de conexão Fonte: autores

A Figura 4 mostra a tela principal do sistema, onde o usuário escolhe o BD dentro do seu servidor, e ao selecionar, exibirá o BD na sua forma relacional, ao clica no ícone dos grafos, o BD será convertido da sua forma orientado a grafos. E depois o usuário poderá salvar a imagem de representação do grafo.

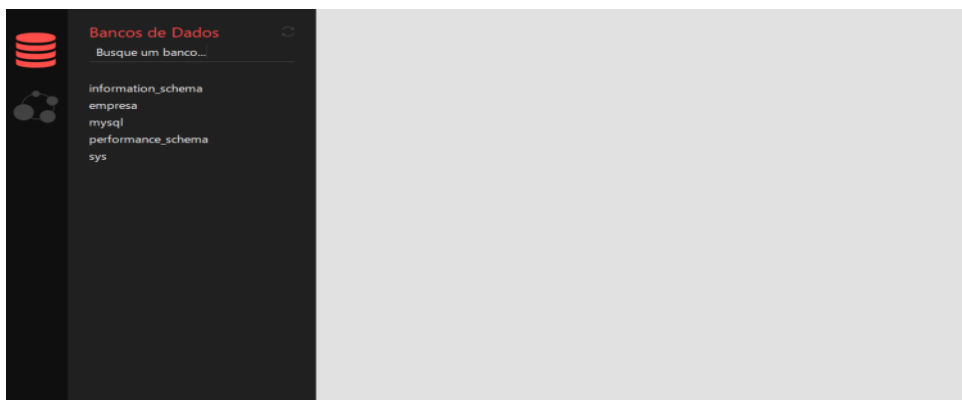


Figura 4 - Tela Principal Fonte: Autores

Na tela (figura 5) mostra a conversão de um banco de dados feita pelo *software*. Foi selecionado o banco de dados e em seguida foi convertido para grafos, na figura 6 o grafo aparece no console.

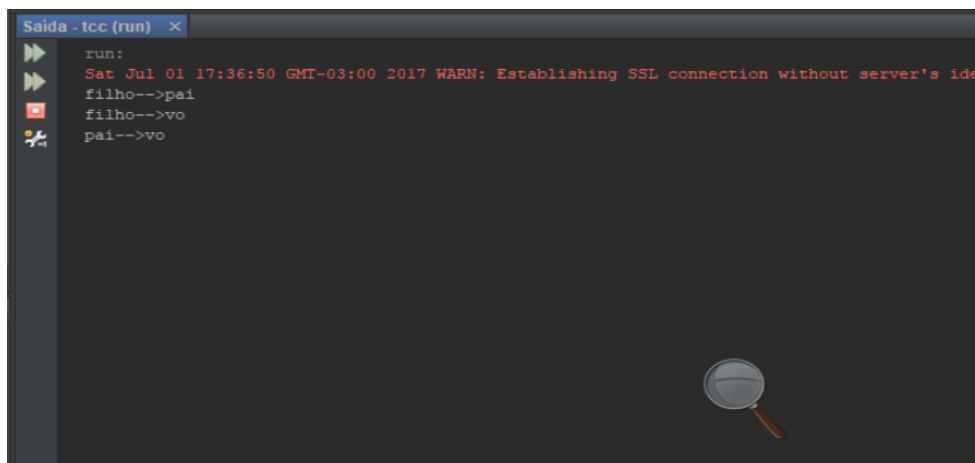


Figura 5 – Banco Convertido Fonte: Autores

Os resultados parciais presente se mostraram importante na continuação do desenvolvimento do *software*, seguindo todo o planejamento e atendendo todos os requisitos que foram instalados para seu desenvolvimento.

5 CONSIDERAÇÕES FINAIS

Esse artigo apresentou o processo de desenvolvimento do sistema de *software* que faz a conversão de banco de dados, que converte uma base de dados na forma relacional para a forma a grafos e que será exibido ao usuário em forma de grafos, que ainda há algumas funções em desenvolvimento.

Durante o processo de desenvolvimento desse sistema de conversão, foi observado a dificuldade no uso de novas tecnologias que foram utilizadas no desenvolvimento do sistema, como a API JGraph responsável pela representação gráfica dos grafos. Essas dificuldades contribuíram na ampliação do conhecimento sobre as linguagens Java, SQL e UML que estão usadas no desenvolvimento do sistema, é sua devido utilização no âmbito da programação.

No momento algumas funções do sistema já estão finalizadas, sendo elas, a conexão com o servidor onde estão presentes o banco de dados do usuário, algumas telas do sistema como a de conexão com o servidor, tela principal do sistema onde estará presente todas as funções, e o algoritmo de conversão do sistema.

A próxima etapa será concluir e ampliar as funções do sistema já existentes e outras que estão em pleno desenvolvimento. Será utilizada a API JGraph na, criação gráfica dos grafos e suas possíveis representações no sistema, Ampliação das telas do sistema, entre outras. No futuro será adicionado novas funções, como uma ampliação no sistema de busca dos bancos de dados,

visualização de bancos já convertidos direto no sistema, um melhor *layout* das telas e funções de personalização de sistema feitas pelo próprio usuário, podendo ajustar da sua forma e preferência.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Adriano. **Trabalhando com Relacionamentos: bancos de dados baseados em grafos e o Neo4j**. 2011. Disponível em: <<http://blog.caelum.com.br/trabalhando-com-relacionamentos-bancos-de-dados-baseados-em-grafos-e-o-neo4j/>>. Acesso em: 25 fev. 2017.

ALVAREZ, Guilherme M.; CECI, Flávio; GONÇALVES, Alexandre L. **Análise Comparativa dos Bancos Orientados a Grafos de Primeira e Segunda Geração: Uma Aplicação na Análise Social**. Florianópolis, 2016. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/eise/2016/003.pdf>>. Acesso em: 04 jun. 2017.

CARVALHO, Marco Antônio Garcia de. **Teoria dos Grafos - Uma Introdução**. São Paulo: Ceset, 2005. Disponível em: <http://www.ft.unicamp.br/~magic/ft024/apografos_ceset_magic.pdf>. Acesso em: 25 mar. 2017.

CAVOTO, Patrícia Raia Nogueira. **Arquitetura Híbrida de Integração entre Banco de Dados Relacional e de Grafos: Uma Abordagem Aplicada à Biodiversidade**. Curitiba, 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wtdbd/2014/003.pdf>>. Acesso em: 04 de junho de 2017.

CÉSAR, Júlio. **O que é SQL e qual sua importância?** São Paulo: Trocando Ideias, 2012. Disponível em: <<https://trocandoideias.wordpress.com/2012/01/27/o-que-e-sql-e-qual-sua-importancia/>>. Acesso em: 28 de maio de 2017.

DIANA, Mauricio de; GEROSA, Marco Aurélio. **NOSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Relacionais para Armazenamento de Dados na Web 2.0. IX Workshop de Teses e Dissertações em Banco de Dados**, Belo Horizonte, out. 2010. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wtdbd/2010/sbbd_wtd_12.pdf>. Acesso em: 16 mar. 2017.

GUEDES, Gilleanes T. A. **UML 2: Uma abordagem prática**. 2. ed. São Paulo: Novatec, 2011. 488 p. Disponível em: <<https://s3.novatec.com.br/capitulos/capitulo-9788575222812.pdf>> Acesso em: 27 de maio de 2017.

JAVA. **Obtenha Informações sobre a Tecnologia Java**. Disponível em: <https://www.java.com/pt_BR/about/> Acesso em: 24 de maio de 2017.

JGRAPH. Disponível em:

<<https://www.jgraph.com/downloads/jgraph/api/org/jgraph/JGraph.html>> Acesso em: 28 de maio de 2017.



NETBEANS. Disponível em: <<https://netbeans.org/features/index.html>> Acesso em: 27 de maio de 2017.

RODRIGUES, Helder. **As Pontes de Königsberg**. Disponível em: <<http://www.helderrodrigues.eu/wp-content/images/konigsberg.jpg>> Acesso em 21 mar. 2017.

VAINOLO. **Interface do JGraph**. Disponível em: <<https://www.vainolo.com/2011/04/11/another-day-with-jgraph-styles-and-constrained-children/>> Acesso em 18 de junho de 2017.