

Armazenando Dados em Aplicações Java

Parte 1 de 3: Entendendo o problema

Hua Lin Chang Costa, hualin@cos.ufrj.br, COPPE/UFRJ.

Leonardo Gresta Paulino Murta, leomurta@ic.uff.br, IC/UFF.

Vanessa Braganholo, braganholo@dcc.ufrj.br, DCC/UFRJ.

Quais são os problemas e as opções disponíveis para armazenamento de dados em aplicações Java de verdade? Este artigo é o primeiro de uma trilogia, e responde à parte “quais são os problemas” dessa pergunta.

Quando estamos em um ambiente acadêmico e precisamos fazer um programa para alguma disciplina ou trabalho final de curso, o armazenamento dos dados é de pouca importância, e soluções como serialização normalmente são suficientes. Contudo, depois de formados precisaremos lidar com sistemas reais, em ambientes reais, e a solução anterior deixa de ser satisfatória. Este artigo é o primeiro de uma trilogia, e apresenta um caso real para ilustrar quais são os principais problemas relacionados ao armazenamento de dados em aplicações Java.

Para podermos entender os problemas reais em relação ao armazenamento de dados em aplicações Java, escolhemos um sistema em desenvolvimento há mais de 10 anos: o *Ambiente Odyssey*. Esse sistema consiste em uma ferramenta CASE (*Computer-Aided Software Engineering*) com ênfase em reutilização de software e engloba funcionalidades como diagramação UML (*Unified Modeling Language*), diagramação voltada para reutilização, documentação de artefatos, geração automática de componentes de negócio, geração de código, entre outras. A sua primeira versão data de 1998, praticamente dois anos após o surgimento da linguagem Java. O Ambiente Odyssey é um projeto acadêmico bastante ativo, desenvolvido pelo grupo de Reutilização de Software da COPPE/UFRJ, que já contou com diversos mecanismos de armazenamento de dados, e a sua equipe já vivenciou na prática os problemas relacionados a esse tema.

Esse artigo inicia apresentando os mecanismos de armazenamento de dados que o Ambiente Odyssey já utilizou nos últimos 10 anos. Em seguida, apresenta os pontos positivos e negativos de cada um desses mecanismos e encerra levantando os principais requisitos relacionados ao armazenamento de dados que puderam ser observados a partir desse caso real. O objetivo, ao investigar as características desses mecanismos, é mostrar como é amplo o horizonte de soluções em persistência e como essas soluções são diferentes entre si.

Analisando um caso: o Ambiente Odyssey

Diferentes mecanismos de armazenamento foram incorporados ao Ambiente Odyssey ao longo dos anos. As tecnologias utilizadas e a abordagem de cada mecanismo se diferenciam principalmente em virtude do contexto da época em que foram implantadas. Até o momento três soluções de armazenamento já foram implantadas para o Ambiente Odyssey: o GOA++, o MOR e Serialização.

O GOA++ é um gerente de objetos armazenados com suporte a representação de objetos diretamente em disco. Desenvolvido também como um projeto acadêmico pelo grupo de Banco de Dados da COPPE/UFRJ, o GOA++ possui uma arquitetura aberta e flexível, e essas características foram determinantes para sua adoção ao invés de outros produtos do mercado. O GOA++ inicialmente atendeu às necessidades do ambiente Odyssey, mas com o aumento da complexidade do projeto algumas dificuldades surgiram, como será apresentado mais adiante.

O MOR é uma abordagem para armazenamento de objetos Java em bancos de dados relacionais de forma transparente. O mapeamento entre objetos Java e o modelo relacional é realizado através do mecanismo de reflexão provido pela linguagem Java. O acesso ao banco de dados relacional é feito

através da ponte JDBC (*Java DataBase Connectivity*). O MOR surgiu como uma alternativa ao uso de Sistemas Gerenciadores de Banco de Dados (SGBD) orientados a objetos, que não se popularizaram frente ao modelo relacional que apresentava SGBDs mais robustos e com melhor desempenho.

A Serialização é um mecanismo fornecido pela linguagem Java para representação do estado de um objeto através de um fluxo binário, permitindo que ele possa ser armazenado e recuperado. A Serialização é atualmente o principal mecanismo de armazenamento oferecido pelo Odyssey. Por ser um mecanismo oferecido pela própria linguagem Java ele é simples de ser utilizado e realiza o processo de armazenamento de forma transparente e automática.

Pesquisa para compreender o problema

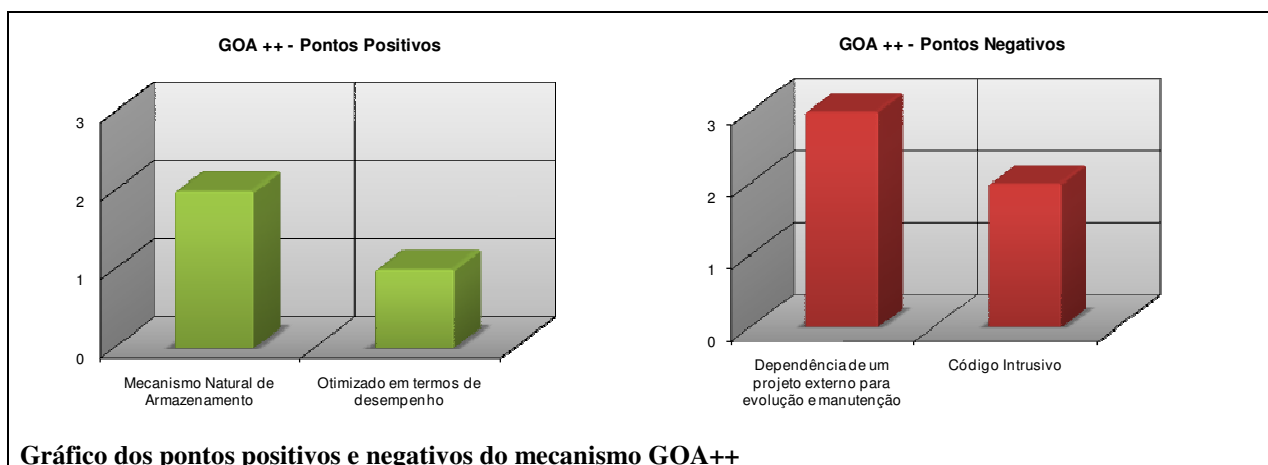
As características do Ambiente Odyssey o tornam um caso propício para entendermos quais são os reais problemas de armazenamento de dados em Java. Para isso, foi realizada uma pesquisa com os desenvolvedores e colaboradores do projeto. A partir das experiências de cada um dos participantes foi possível avaliar os pontos positivos e negativos de cada um dos mecanismos oferecidos pelo Odyssey durante os últimos 10 anos. Além disso, foram coletados também os requisitos de um mecanismo de armazenamento ideal a partir do ponto de vista de cada envolvido e de sua experiência com o Ambiente Odyssey.

A pesquisa teve a colaboração, no total, de quatorze participantes: dois alunos de Iniciação Científica, sete alunos de Mestrado e cinco alunos de Doutorado. As diferentes partes da pesquisa tiveram um número variante de participantes em função da dificuldade em contatar as pessoas que tiveram experiência com os mecanismos adotados no início do projeto, como o GOA++.

Resultados obtidos: GOA++

Dentre os 14 participantes da pesquisa, somente três contribuíram com a avaliação do mecanismo GOA++. Esse fato pode ser explicado em virtude desse mecanismo ser o mais antigo dos três. Sendo assim, poucas pessoas que tiveram experiência com ele puderam ser contatados para a pesquisa.

A figura a seguir mostra os pontos positivos e negativos do GOA++ relatados pelos participantes. O gráfico relaciona cada característica citada com o número de vezes que ela foi mencionada pelos participantes. Os pontos citados pelos participantes são em virtude de suas experiências com o mecanismo GOA++.



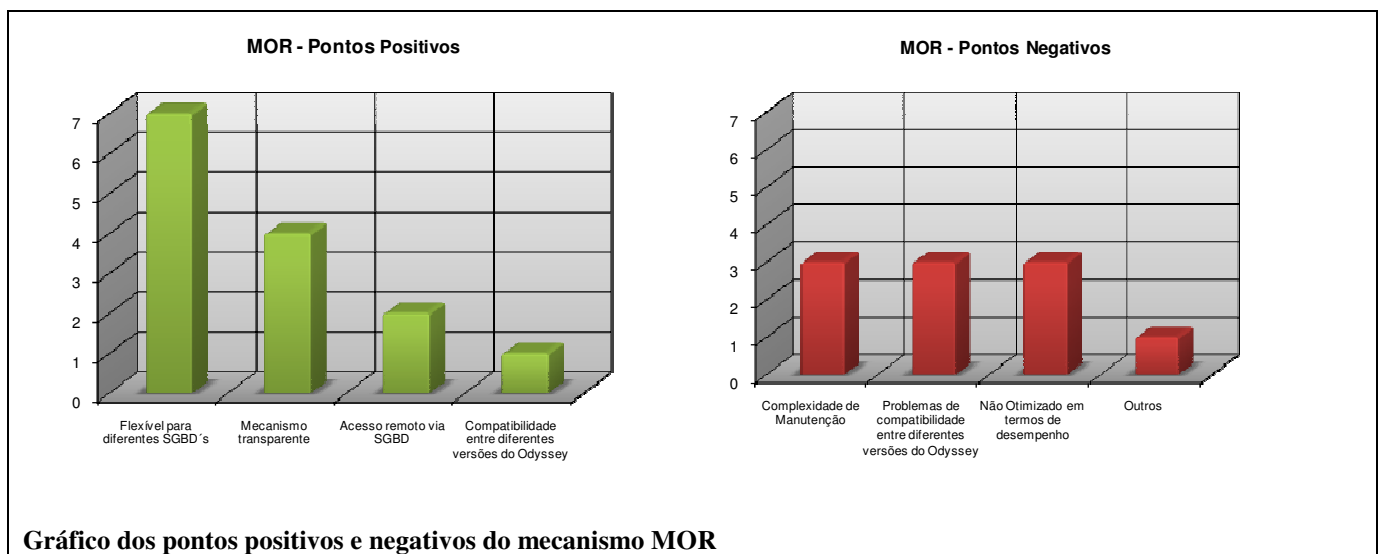
O GOA++ não é um produto fechado e isso gera características positivas, como, por exemplo, sua arquitetura aberta e flexível. Mas isso acaba gerando também características desfavoráveis à sua adoção, como é o caso do ponto negativo citado por todos os participantes com experiência no mecanismo: a dependência de um projeto externo para evolução e manutenção. O GOA++ é um projeto voltado para realidades acadêmicas e, em virtude disso, é um protótipo que pode apresentar instabilidades. A dependência do GOA++ para manutenção do mecanismo de armazenamento do

Odyssey é um problema grave e muito citado pelos participantes da pesquisa. É importante notar que os pontos negativos citados pelos participantes refletem justamente os pontos que influenciaram na decisão de substituição deste mecanismo de persistência.

Resultados obtidos: MOR

Dentre os 14 participantes da pesquisa, oito deles relataram experiência com o mecanismo MOR. O número maior de participantes, em relação ao mecanismo GOA++, pode ser explicado pelo fato do MOR ser um mecanismo mais recente.

A figura a seguir mostra os pontos positivos e negativos citados pelos participantes da pesquisa em relação ao mecanismo MOR. Pelo fato de o MOR ser um mecanismo que envolve o processo de mapeamento objeto-relacional, os pontos positivos e negativos basicamente envolvem os prós e contras dessa abordagem.



Dentre os pontos positivos mais citados pelos participantes, dois deles se destacam. Primeiro, o fato de o mecanismo ser flexível para diferentes SGBDs, que é proporcionado pelo uso da ponte JDBC para conexão com o banco de dados. Segundo, a transparência, alcançada pelo uso de reflexão da linguagem Java para automatização do mapeamento objeto-relacional.

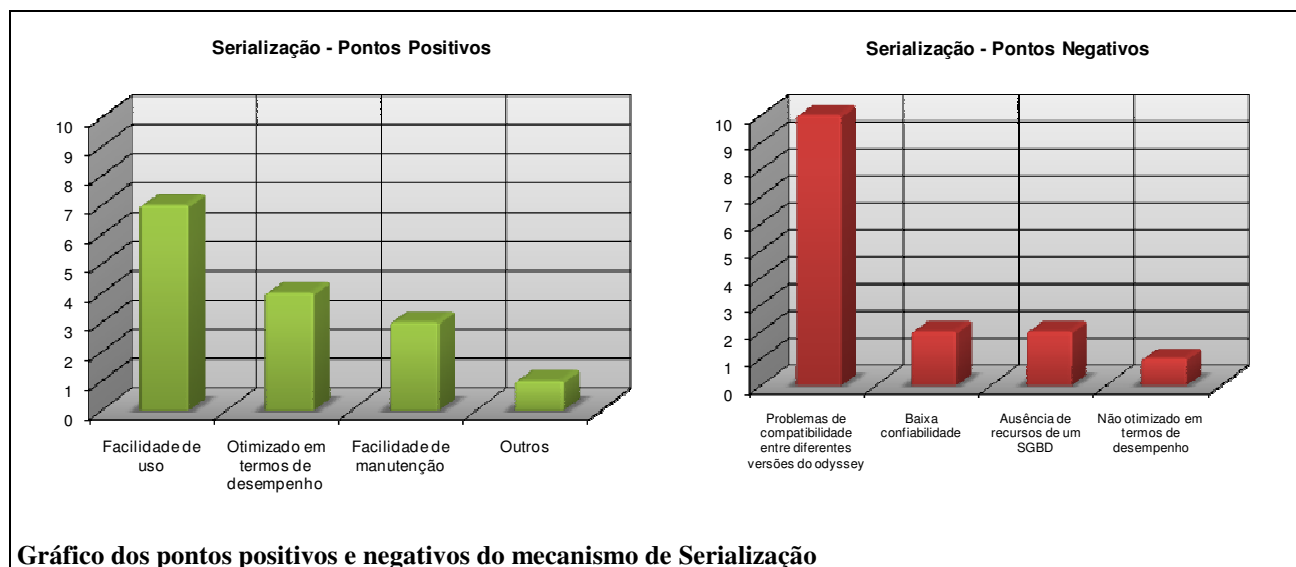
Apesar dos mecanismos citados acima, ponte JDBC e reflexão, facilitarem o processo de mapeamento objeto-relacional, eles são mecanismos complexos. A utilização deles implica em um aumento no grau de complexidade na manutenção do projeto. Essa complexidade é justamente um dos pontos negativos mais citados pelos participantes em relação ao mecanismo MOR.

Resultados obtidos: Serialização

Dentre os três mecanismos de armazenamento avaliados, a Serialização apresentou o maior número de participantes, um total de treze. Isso acontece pelo fato de Serialização ser o principal mecanismo de armazenamento oferecido pelo Odyssey atualmente.

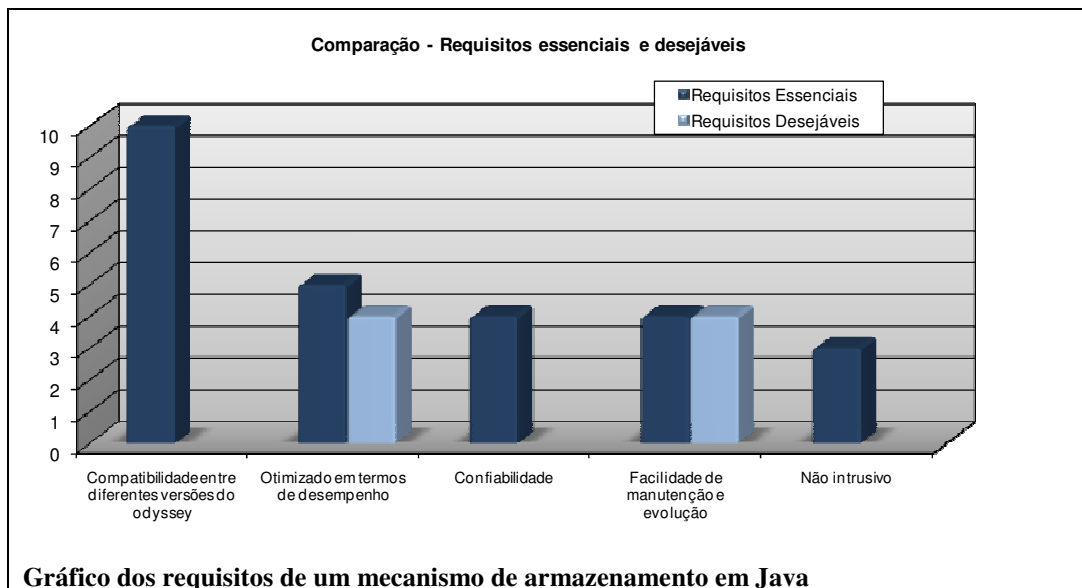
Na figura a seguir é possível ver os pontos positivos e negativos mencionados pelos participantes em relação à Serialização. Os pontos positivos mais citados pelos participantes envolvem as vantagens já citadas do próprio mecanismo de Serialização: facilidade de uso e de manutenção.

O ponto negativo citado pela maioria dos participantes da pesquisa com experiência com a Serialização foi o problema de compatibilidade entre diferentes versões do Odyssey. Ou seja, não ser possível acessar em uma versão mais recente os dados gravados por versões mais antigas. Essa é a deficiência mais amplamente observada nos resultados da avaliação da Serialização, mas está presente em todos os três mecanismos de armazenamento.



Requisitos para armazenamento de dados em Java

A figura abaixo mostra os principais requisitos coletados pela pesquisa. A pesquisa, além de coletar os requisitos, pediu para que os participantes os classificassem como desejáveis ou essenciais. É importante notar que os requisitos citados pelos participantes refletem de maneira coerente os pontos positivos e negativos dos mecanismos citados anteriormente. Ou seja, é natural que alguém com experiência com um dos três mecanismos de armazenamento suportados pelo Odyssey deseje um novo mecanismo que solucione as deficiências (pontos negativos) e, na medida do possível, mantenha as características positivas dos mecanismos anteriores.



Concluindo

Tendo em vista os problemas encontrados nas tecnologias discutidas e os principais requisitos coletados pela pesquisa, o próximo passo é buscar soluções que atendam da melhor forma possível essas características. No próximo artigo da trilogia, veremos quais são as tecnologias disponíveis para tratar o problema de armazenamento de dados em Java e quais são as características principais dessas tecnologias.

Recursos

Ambiente Odyssey: <http://reuse.cos.ufrj.br/odyssey>

GOA++: Mattoso, M., Werner, C. M. L., Braga, R. M. M., Pinheiro, R., Murta, L. G. P., Almeida, V., Costa, M., Bezerra, E., Soares, J., Ruberg, N.: Persistência de Componentes num Ambiente de Reuso. Simpósio Brasileiro de Engenharia de Software (SBES), Sessão de Ferramentas, João Pessoa, outubro (2000) 351-354

MOR: Murta, L. G. P., Veronese, G. O., Werner, C. M. L.: MOR: Uma Ferramenta para o Mapeamento Objeto-Relacional em Java. Simpósio Brasileiro de Engenharia de Software (SBES), Sessão de Ferramentas, Rio de Janeiro, outubro (2001) 392-397

Serialização: <http://java.sun.com/javase/6/docs/technotes/guides/serialization/index.html>

Sobre os autores



Hua Lin Costa é aluno de mestrado do Programa de Engenharia de Sistemas e Computação (COPPE/UFRJ) na linha de pesquisa de Engenharia de Software. Possui graduação em Ciência da Computação (2008) pela UFRJ. Atualmente realiza trabalhos acadêmicos junto ao Grupo de Reutilização de Software da COPPE como colaborador no desenvolvimento do ambiente Odyssey e do Projeto Brechó.



Leonardo Murta é sócio da SBC e professor de Engenharia de Software do Instituto de Computação da Universidade Federal Fluminense. Possui graduação em Informática (1999) pela UFRJ e mestrado (2002) e doutorado (2006) em Engenharia de Sistemas e Computação, também pela UFRJ. Suas principais áreas de interesse são gerência de configuração, reutilização e arquiteturas e software. Mais informações podem ser obtidas em <http://www.ic.uff.br/~leomurta>.



Vanessa Braganholo é sócia da SBC e professora do Departamento de Ciência da Computação do Instituto de Matemática da Universidade Federal do Rio de Janeiro. Possui graduação em Ciência da Computação (1998) pela Universidade Federal do Rio Grande do Sul e doutorado (2004) em Ciência da Computação também pela UFRGS. Tem atuado em diversos projetos de pesquisa ligados a dados semi-estruturados. Sua principal área de atuação é bancos de dados. Mais informações podem ser obtidas em <http://www.dcc.ufrj.br/~braganholo>.